

信息安全理论与技术系列丛书

丛书主编：冯登国

国家重点基础研究发展规划项目资助（项目编号：2007CB311202）
国家自然科学基金重点项目资助（项目编号：60833008）

信息安全中的数学方法与技术

冯登国　等　编著



清华大学出版社

信 息 安 全 理 论 与 技 术 系 列 丛 书

信息安全中的数学方法与技术

冯登国 等　编著

清 华 大 学 出 版 社

北　京

内 容 简 介

本书主要介绍了研究和掌握信息安全理论与技术必备的数学方法与技术，主要内容包括初等数论、代数、椭圆曲线、组合论、图论、概率论、信息论、数理统计、随机过程、频谱、纠错编码、计算复杂性、数理逻辑、数字信号处理、数据挖掘等方法与技术，并同步介绍了这些方法与技术在信息安全中的典型应用。

本书可作为高等院校信息安全、密码学、数学、计算机、通信等专业的博士生、硕士生和本科生的教科书，也可供从事相关专业的教学、科研和工程技术人员参考。

图书在版编目（CIP）数据

信息安全中的数学方法与技术 / 冯登国等编著. —北京：清华大学出版社，2009.10
（信息安全理论与技术系列丛书）
ISBN 978-7-302-20966-9

Ⅰ. 信…　Ⅱ. 冯…　Ⅲ. 信息系统—安全技术—应用数学　Ⅳ. TP309　O29

中国版本图书馆 CIP 数据核字(2009)第 164548 号

责任编辑：张　民　张为民
责任校对：李建庄
责任印制：何　芊

出版发行：清华大学出版社　　**地　址**：北京清华大学学研大厦 A 座
http://www.tup.com.cn　　**邮　编**：100084
社 总 机：010-62770175　　**邮　购**：010-62786544
投稿与读者服务：010-62776969，c-service@tup.tsinghua.edu.cn
质量反馈：010-62772015，zhiliang@tup.tsinghua.edu.cn
印 刷 者：北京鑫海金澳胶印有限公司
装 订 者：三河市新茂装订有限公司
经　销：全国新华书店
开　本：185×260　　**印　张**：28.5　　**字　数**：654 千字
版　次：2009 年 10 月第 1 版　　**印　次**：2009 年 10 月第 1 次印刷
印　数：1～3000
定　价：49.00 元

本书如存在文字不清、漏印、缺页、倒页、脱页等印装质量问题，请与清华大学出版社出版部联系调换。联系电话：010-62770177 转 3103　　产品编号：032959-01

丛书序

Preface

信息安全已成为国家安全的重要组成部分，也是保障信息社会和信息技术可持续发展的核心基础。信息技术的迅猛发展和深度应用必将带来更多难以解决的信息安全问题，只有掌握了信息安全的科学发展规律，才有可能解决人类社会遇到的各种信息安全问题。但科学规律的掌握非一朝一夕之功，治水、训火、利用核能曾经都经历了多么漫长的岁月。

无数事实证明，人类是有能力发现规律和认识真理的。今天对信息安全的认识，就经历了一个从保密到保护，又发展到保障的趋于真理的发展过程。信息安全是动态发展的，只有相对安全没有绝对安全，任何人都不能宣称自己对信息安全的认识达到终极。国内外学者已出版了大量的信息安全著作，我和我所领导的团队近10年来也出版了一批信息安全著作，目的是不断提升对信息安全的认识水平。我相信有了这些基础和积累，一定能够推出更高质量和更高认识水平的信息安全著作，也必将为推动我国信息安全理论与技术的创新研究做出实质性贡献。

本丛书的目标是推出系列具有特色和创新的信息安全理论与技术著作，我们的原则是成熟一本出版一本，不求数量，只求质量。希望每一本书都能提升读者对相关领域的认识水平，也希望每一本书都能成为经典范本。

我非常感谢清华大学出版社给我们提供了这样一个大舞台，使我们能够实施我们的计划和理想，我也特别感谢清华大学出版社张民老师的支持和帮助。

限于作者的水平，本丛书难免存在不足之处，敬请读者批评指正。

冯登国

2009年夏于北京

冯登国，中国科学院软件所研究员，博士生导师，教育部高等学校信息安全类专业教学指导委员会副主任委员，国家信息化专家咨询委员会专家，国家863计划信息安全技术主题专家组组长，信息安全国家重点实验室主任，国家计算机网络入侵防范中心主任。

前言

foreword

信息安全作为一门重要的学科方向，与其他学科一样，有其自身的方法论。从理论与技术研究角度来看，信息安全有其自身的研究方法学；从管理角度来看，信息安全有其自身的管理方法学；从工程与应用角度来看，信息安全有其自身的工程方法学。本书重点讲述信息安全的研究方法学，我们称之为信息安全中的数学方法与技术。数学方法与技术是研究和掌握信息安全理论与技术的基础和工具。

面向信息安全专业本科生教育的数学教材《信息安全数学基础》是从基础的角度介绍与信息安全相关的数学基础知识，本书则是从研究与打基础并重的角度介绍研究和掌握信息安全理论与技术必备的数学方法与技术。本书的特点如下：

(1) 内容全面。涵盖了当前研究信息安全理论与技术的主要方法与技术，包括初等数论、代数、椭圆曲线、组合论、图论、概率论、信息论、数理统计、随机过程、频谱、纠错编码、计算复杂性、数理逻辑、数字信号处理、数据挖掘、软件安全性分析等方法与技术。

(2) 针对性强。紧密结合信息安全理论与技术研究的需求和掌握信息安全理论与技术工具的需求，重点介绍研究方法与技术，并选择有代表性的应用进行举例，将研究方法与技术和信息安全融为一体。不仅适用于专门从事信息安全研究的专业人员，而且也适用于从事相关理论与技术的研究人员了解理论与技术在信息安全中的应用示范。

(3) 起点高。重点从研究的视角介绍信息安全中的数学方法与技术，并对方法和技术做了高度提炼。例如，纠错编码方法与技术这一章，不仅是对信息安全研究中所用到的纠错编码方法与技术的高度总结，而且也是现有纠错编码重要方法与技术的一个高度概括。

本书是作者长期从事信息安全研究工作的方法和经验的总结，同时，也吸收了国内外现有相关著作中的一些精华，这些相关著作已在参考文献中列出。另外，本书在中国科学院研究生院开设的研究生课程中和信息安全国家重点实验室的研究生班中讲授过多次，这些实践工作对本书的形成具有十分重要的意义。

本书是由在一线从事信息安全研究的科研工作者完成的。第 1 章由胡磊教授执笔，第 2 章由林东岱研究员执笔，第 3 章由王鲲鹏副教授执笔，第 4 章由周林芳副教授执笔，第 5 章的 5.1 节～5.5 节由李宝教授执笔，5.6 节～5.7 节由冯登国教授执笔，第 6 章由张振峰副研究员和冯登国教授执笔，第 7 章由陈华副研究员和冯登国教

授执笔，第 8 章由吴文玲研究员执笔，第 9 章由武传坤研究员执笔，第 10 章由赵亚群教授和冯登国教授执笔，第 11 章由冯登国教授执笔，第 12 章由徐静副研究员执笔，第 13 章由薛锐研究员执笔，第 14 章由赵险峰副研究员和冯登国教授执笔，第 15 章由连一峰副研究员和冯登国教授执笔，第 16 章由苏璞睿副研究员和冯登国教授执笔。全书由冯登国教授策划和统稿。

本书在写作过程中得到了清华大学出版社的大力支持和国家重点基础研究发展规划项目（项目编号：2007CB311202）和国家自然科学基金重点项目（项目编号：60833008）的资助，在此表示衷心的感谢。

冯登国

2009 年 8 月于北京

目录

Contents

第 1 章　初等数论方法与技术

数论是研究自然数 1、2、3、…性质的一门数学分支。历史上，人们很早就开始数论的研究，取得了十分丰富的研究成果。今天，数论不仅自身发展成为一门浩瀚、艰深的数学分支，而且已成为其他领域(如通信、信息安全)的重要研究工具。

信息安全中涉及很多数论方法与技术，限于篇幅，本章重点介绍信息安全中常用的一些数论结论和数论算法，包括中国剩余定理、欧拉定理、欧氏算法、素数检测算法等。最后，作为这些数论结果的应用，介绍 RSA 密码算法和 Rabin 密码算法的工作原理。

1.1　基本概念

本节主要介绍初等数论中的一些基本概念，包括整除、同余、剩余类、二次剩余等。

1.1.1　整除

通常用$\mathbb{Z}$表示整数的集合，即

$$\mathbb{Z}=\{0,\pm 1,\pm 2,\cdots\}$$

众所周知，$\mathbb{Z}$ 对于普通的加、减、乘 3 种运算是封闭的，即$\mathbb{Z}$ 中的任何两个数的加、减、乘的运算结果都在$\mathbb{Z}$ 中，但$\mathbb{Z}$ 对于除法是不封闭的。给定两个整数 a 和 b，$b>0$，则有所谓的带余除法，即存在整数 q 和 r，使得

$$a=qb+r,\quad 0\leqslant r<b \tag{1.1}$$

通常将 q 称为 a 除以 b 的商，r 称为 a 除以 b 的余数(又称最小非负剩余)。

事实上，令 $q=\left\lfloor\frac{a}{b}\right\rfloor$ 是不超过分数$\frac{a}{b}$的最大整数，则 $0\leqslant\frac{a}{b}-q<1$，即有

$$0\leqslant a-qb<b$$

令 $r=a-qb$，则 q、r 满足式(1.1)的要求。

当 $r=0$ 时，$a=qb$，则称 b 整除 a，或 b 是 a 的因子或 a 是 b 的倍数，记做 $b\mid a$；反之，若上述最小非负剩余 $r\neq 0$，则称 b 不整除 a，记做 $b\nmid a$。

易验证，整除有以下性质：

(1) 若 $a\mid b$，$b\mid c$，则 $a\mid c$；

(2) 若 $a\mid b$，$a\mid c$，则 $a\mid b\pm c$；

(3) 若 $a\mid b_i$，$c_i\in\mathbb{Z}$，$i=1,2,\cdots,m$，则 $a\,\Big|\,\sum\limits_{i=1}^{m}c_ib_i$。

整除的概念自然可以推广到因子是负整数的情形。不难验证，若 $a\mid b$，$b\mid a$，则 $a=\pm b$。

1.1.2 最大公因子

设 a 和 b 是不全为 0 的整数，能够同时整除 a 和 b 的最大正整数 d，称为 a 和 b 的最大公因子，记做 $d=\gcd(a,b)$。

最大公因子一定存在。事实上，a 和 b 的最大公因子存在于集合

$$S=\{ma+nb \mid m,n\in\mathbb{Z}\}$$

之中，并且是这个集合中的最小正整数。记 S 中的最小正整数为 $d=m'a+n'b$，可以证明 $d=\gcd(a,b)$。首先，这个集合中的每个元素 $ma+nb$ 一定是 d 的倍数。作带余除法，$ma+nb=dq+r$，则

$$r=ma+nb-q(m'a+n'b)=(m-qm')a+(n-qn')b\in S$$

因为 d 是 S 中的最小正整数，而 $0\leqslant r<d$，所以 $r=0$，即 $d\mid ma+nb$。其次 $d\mid a$，因为 $a=1\cdot a+0\cdot b\in S$。同理 $d\mid b$，所以 d 是 a 和 b 的公因子。再次，对于 a 和 b 的任意一个公因子 d'，因为 $d'\mid a$，$d'\mid b$，所以 d' 整除 $m'a+n'b$，即 $d'\mid d$。因此 d 是 a 和 b 的最大公因子。

上述证明过程表明，最大公因子 $d=\gcd(a,b)$ 满足以下性质：

(1) 存在正整数 m 和 n，使得 $d=ma+nb$；

(2) 对任意整数 m 和 n，均有 $d\mid ma+nb$；

(3) 若 $c\mid a$，$c\mid b$，则 $c\mid d$。

另外，从最大公因子的定义出发可以直接证明，对任意整数 a、b、c，都有 $\gcd(a,b)=\gcd(a+bc,b)$。

若两个整数的最大公因子为 1，则称它们互素。

一个大于 1 的整数 p 称为素数，如果它只有 1 和 p 两个正的因子。对于素数 p，则有以下性质：若 $p\mid ab$，则 $p\mid a$ 或 $p\mid b$。

这是因为若 $p\nmid a$，则 $\gcd(p,a)=1$。设 $m,n\in\mathbb{Z}$ 使得 $am+pn=1$，则

$$b=(ab)m+p(nb)$$

是 p 的倍数，因为上式右边的 ab 和 p 都是 p 的倍数。

这个性质可以推广为：若 $\gcd(a,m)=1$，$m\mid ab$，则 $m\mid b$。

1.1.3 同余式

如果两个整数 a 和 b 除以 m 的最小非负剩余(余数)是相同的，则称它们模 m 同余，记做

$$a\equiv b\pmod{m}$$

显然，a 和 b 模 m 同余等价于 $m\mid a-b$。

不难证明，同余具有以下性质：

(1) 若 $a_i\equiv b_i\pmod{m}$，$i=1,2$，则 $a_1+a_2\equiv b_1+b_2\pmod{m}$，$a_1a_2\equiv b_1b_2\pmod{m}$；

(2) 若 $ac\equiv bc\pmod{m}$，$\gcd(c,m)=1$，则 $a\equiv b\pmod{m}$；

(3) 若 $a\equiv b\pmod{m}$，d 是 a,b,m 的公因子，则 $\frac{a}{d}\equiv\frac{b}{d}\left(\bmod \frac{m}{d}\right)$。

1.1.4　剩余类

整数相除时大多数情形是最小非负剩余(余数)不为 0。依照余数,可以将整数集合$\mathbb{Z}$划分成若干个子集。

设 m 是正整数,$0\leqslant i\leqslant m-1$。将除以 m 余 i 的所有整数形成的集合记做 C_i,则

$$C_i=\{\cdots,-2m+i,-m+i,i,m+i,2m+i,\cdots\}$$

显然,整数集$\mathbb{Z}$被划分成 $C_0,C_1,\cdots,C_{m-1}$ 这 m 个子集合,或者说$\mathbb{Z}$是这 m 个不相交的子集的并。这 m 个子集称为模 m 的 m 个剩余类。

显然,a 和 b 属于模 m 的同一个剩余类,等价于 a 和 b 模 m 同余。

剩余类的一个特点是:可以在剩余类形成的集合上定义加法和乘法运算,这使得剩余类的集合成为一个拥有加、减、乘 3 种运算的代数集合,而这个代数集是一个有限集合,只包含 m 个元素(即 m 个剩余类)。与无限代数集如$\mathbb{Z}$不同的是,有限代数集合更适合应用于信息处理的许多领域,如通信、信息安全等。

剩余类的加法和乘法具体定义如下:

$$C_i+C_j=\{a+b\mid a\in C_i,b\in C_j\}$$

$$C_i\cdot C_j=\{a\cdot b\mid a\in C_i,b\in C_j\}$$

记$\mathbb{Z}_m$是模 m 的剩余类 $C_0,C_1,\cdots,C_{m-1}$ 的集合。不难看出,由于剩余类有以下性质:

$$C_i=i+C_0$$

即 C_i 中每个元素是 i 与零剩余类 C_0 中某个元素之和(反之亦然),上述剩余类的加法和乘法实际上是

$$C_i+C_j=i+j+C_0=C_u$$

$$C_i\cdot C_j=i\cdot j+C_0=C_v$$

其中 u 和 v 分别是 $i+j$ 和 $i\cdot j$ 除以 m 的最小非负剩余。因此,$\mathbb{Z}_m$的代数运算实际上是 i 和 j 的相应运算,再将结果对应到某个剩余类。如果将剩余类用 $0,1,\cdots,m-1$ 来标记,剩余类 i 和 j 的运算等价于将整数 i 和 j 的运算结果除以 m 求最小非负剩余。实际上在每个剩余类 C_i 中选一个代表元,剩余类的运算实际上就是它们的代表元的运算。m 个剩余类的 m 个代表元(每个代表元对应一个不同的剩余类)构成一个模 m 完全剩余系。可将$\mathbb{Z}_m$等同于任何一个模 m 完全剩余系,如$\mathbb{Z}_m=\{0,1,\cdots,m-1\}$。

显然,剩余类的代表不是唯一的,一个剩余类中的每个元素都可以选作代表。

1.1.5　欧拉函数与既约剩余系

在模 m 的一个剩余类中,若有一个数与 m 互素,则该剩余类中所有的元素都与 m 互素。称该剩余类为既约剩余类。它们是代表元与 m 互素的剩余类。

模 m 既约剩余类的个数记为 $\varphi(m)$。称 $\varphi(\cdot)$为欧拉函数。显然,$\varphi(m)$等于 $0,1,\cdots,m-1$ 中与 m 互素的整数的个数。在每个既约剩余类中取出一个代表元,共 $\varphi(m)$个元素,它们组成模 m 的一个既约剩余系。

例 1.1.1　设 $m=15$,$\{1,2,4,7,8,11,13,14\}$是模 15 的一个既约剩余系,$\varphi(15)=8$。

按照剩余类的乘法(即$\mathbb{Z}_m$的乘法),既约剩余类在该运算下封闭,即若 C_i 和C_j 是两个既约剩余类,则 $C_i \cdot C_j$ 也是既约剩余类。换句话说,若整数 i 和 j 都与 m 互素,则 $i \cdot j$ 也与 m 互素(1.1.2 小节最后一个性质)。但是,两个既约剩余类之和不一定是既约剩余类,例如,$C_1 + C_{m-1}$ 等于零剩余类,就不是既约剩余类。人们把既约剩余类的集合记做$\mathbb{Z}_m^*$,那么$\mathbb{Z}_m^*$含有 $\varphi(m)$个元素。

$\mathbb{Z}_m^*$的元素可以列举为模 m 的一个既约剩余系。例如,对于素数 $m=p$,$\mathbb{Z}_p^* = \{1,2,\cdots,p-1\}$。

例 1.1.2 设 $p=7$,$\mathbb{Z}_7^*$的乘法表如表 1.1 所示。

表 1.1 $\mathbb{Z}_7^*$的乘法表

	1	2	3	4	5	6
1	1	2	3	4	5	6
2	2	4	6	1	3	5
3	3	6	2	5	1	4
4	4	1	5	2	6	3
5	5	3	1	6	4	2
6	6	5	4	3	2	1

当 $m=p$ 为素数时,每个非零的剩余类 $C_1, C_2, \cdots, C_{p-1}$ 都是既约剩余类。在$\mathbb{Z}_p$中定义除法为乘法的逆运算(除数不为 C_0),则$\mathbb{Z}_p$是拥有与通常四则运算相同的规律(如交换律、结合律和分配律)的运算体系,它实际上是一个含 p 个元素的有限域,关于$\mathbb{Z}_p$的更多讨论将在 1.3.2 小节进行。

例 1.1.3 在美国数字签名算法 DSA 标准中,人们使用了一个大的$\mathbb{Z}_p$,其中 p 是一个二进制展开在 512 位以上的大素数。DSA 标准中的关键运算是$\mathbb{Z}_p$上的运算,即整数模 p 运算。

1.1.6 二次剩余

设 $m>1$ 为整数,n 与 m 互素。若

$$x^2 \equiv n \pmod m \tag{1.2}$$

有解,则称 n 为模 m 的二次剩余;否则称为二次非剩余。

例 1.1.4 设 $m=11$。$(\pm 1)^2, (\pm 2)^2, (\pm 3)^2, (\pm 4)^2, (\pm 5)^2$ 模 11 的余数是 1,4,9,5,3,模 11 的二次剩余为 1,3,4,5,9,二次非剩余为 2,6,7,8,10。

对于奇素数 $m=p$,模 p 的一个既约剩余系为$\left\{\pm 1, \pm 2, \cdots, \pm \frac{p-1}{2}\right\}$。当 x 跑遍一个既约剩余系时,由式(1.2),n 跑遍模 p 的全部二次剩余。设 $1 \leqslant |x_1| < |x_2| \leqslant \frac{p-1}{2}$,不难证明 x_1^2 模 p 不同于 x_2^2。因此$(\pm 1)^2, (\pm 2)^2, \cdots, \left(\pm \frac{p-1}{2}\right)^2$ 是模 p 的全部不同的二次剩余,将所有的$\frac{p-1}{2}$个模 p 的二次非剩余添加进来,恰好构成模 p 的

既约剩余系。

设 $p>2$ 为素数,定义 Legendre 符号:

$$\left(\frac{n}{p}\right)=\begin{cases}0 & 若\ p\mid n\\ 1 & 若\ n\ 是模\ p\ 的二次剩余\\ -1 & 若\ n\ 是模\ p\ 的二次非剩余\end{cases}$$

显然,当 $n\equiv n'(\text{mod } p)$时,$\left(\frac{n}{p}\right)=\left(\frac{n'}{p}\right)$。

1.2　基本原理

1.2.1　中国剩余定理

我国古代有一部著名的数学专著《孙子算经》,书中有这样一个"物不知其数"问题:"今有物不知其数,三三数之剩二,五五数之剩三,七七数之剩二,问物几何?"这就是要求同余方程组

$$\begin{cases}x\equiv 2(\text{mod } 3)\\ x\equiv 3(\text{mod } 5)\\ x\equiv 2(\text{mod } 7)\end{cases}$$

的正整数解。书中给出了这个问题的答案 $x=23$,其所用方法在明朝程大位的《算法统要》中用四句诗描述如下:

三人同行七十稀,五树梨花廿一枝,

七子团圆正月半,除百零五便得知。

意思是,若令

$$m_1=3,\quad M_1=70,\quad m_2=5,\quad M_2=21,\quad m_3=7,\quad M_3=15$$

则

$$x\equiv 2M_1+3M_2+2M_3\equiv 23(\text{mod } 105)$$

上述 M_i 具有以下特性:对 $i\neq j$ 有

$$\begin{cases}M_i\equiv 1(\text{mod } m_i)\\ M_i\equiv 0(\text{mod } m_j)\end{cases}$$

实际上,这给出了以下一般的求解同余方程组的方法。

定理 1.2.1　设 $m_1,m_2,\cdots,m_k$ 是两两互素的正整数,$N=m_1m_2\cdots m_k$,$M_i=\frac{N}{m_i}$,令 M_i'是使得 $M_i'M_i\equiv 1(\text{mod } m_i)$的正整数,则同余方程组

$$\begin{cases}x\equiv a_1(\text{mod } m_1)\\ x\equiv a_2(\text{mod } m_2)\\ \quad\vdots\\ x\equiv a_k(\text{mod } m_k)\end{cases}\tag{1.3}$$

的解是

$$x\equiv M_1'M_1a_1+M_2'M_2a_2+\cdots+M_k'M_ka_k(\text{mod } N)\tag{1.4}$$

该定理的证明很容易。例如，可直接验证式(1.4)给出的解满足式(1.3)第一个方程(记式(1.4)右边的数为 u)：因为 $M_2, M_3, \cdots, M_k$ 都是 m_1 的倍数，所以 $u \equiv M_1' M_1 a_1 \pmod{m_1}$。又因为 $M_1' M_1 \equiv 1 \pmod{m_1}$，所以 $u \equiv M_1' M_1 a_1 \equiv a_1 \pmod{m_1}$。

定理中 M_i' 存在是因为 M_i 与 m_i 互素。实际上，已假设 $m_1, m_2, \cdots, m_k$ 两两互素。用欧氏算法(详见 1.3.1 小节)可以求出 M_i'，因此，该定理给出了求解同余方程的有效解法。上述 M_i 在我国古代数学著作中被称为衍数，M_i' 称为乘率，宋代数学家秦九韶称上述解法为“求一术”。

定理 1.2.1 在我国被称为“孙子定理”，外国文献中称为“中国剩余定理”。这个定理在数论中很重要，它的思想在近代数学中经常被用到。

设 N 为正整数，$N = p_1^{e_1} p_2^{e_2} \cdots p_k^{e_k}$ 是 N 的完全因子分解(其中 $p_1, p_2, \cdots, p_k$ 是互不相同的素数，$e_i \geqslant 1$)。利用中国剩余定理，可以将求 $x \pmod N$ 的问题转化为求 $x \pmod{p_i^{e_i}}$ 的问题，后者可以进一步借助 p-adic 理论转化成模 p 问题进行求解。

1.2.2 欧拉定理和费马小定理

设 m 是正整数，$\varphi(m)$ 是欧拉函数。对与 m 互素的整数 a，有

$$a^{\varphi(m)} \equiv 1 \pmod m \tag{1.5}$$

这个结论称为欧拉定理。

其证明如下：设 $a_1, a_2, \cdots, a_{\varphi(m)}$ 为模 m 的一个既约剩余系，即 $C_{a_1}, C_{a_2}, \cdots, C_{a_{\varphi(m)}}$ 为模 m 的全部既约剩余系。将这些剩余类与 C_a 相乘，得到 $C_{aa_1}, C_{aa_2}, \cdots, C_{aa_{\varphi(m)}}$，后者也两两不等构成一个完整的既约剩余类集合，因此它们的乘积剩余类必相等，即

$$C_{a_1 a_2 \cdots a_{\varphi(m)}} = C_{a^{\varphi(m)} a_1 a_2 \cdots a_{\varphi(m)}}$$

将两边除以 $C_{a_1 a_2 \cdots a_{\varphi(m)}}$ 得

$$C_1 = C_{a^{\varphi(m)}}$$

即得式(1.5)。

欧拉定理有一个特例——费马小定理。设 $m = p$ 是一个素数，对任给的整数 a，都有

$$a^p \equiv a \pmod p \tag{1.6}$$

费马小定理证明如下：若 a 不是 p 的倍数，则 a 和 p 互素，由于 $\varphi(p) = p-1$，由欧拉定理可知，$a^{p-1} \equiv 1 \pmod p$，两边乘以 a 即得式(1.6)。当 a 是 p 的倍数时，式(1.6)两边都是 p 的倍数，因而自然成立。

1.2.3 欧拉函数的计算

在 1.1.5 小节中曾经定义了欧拉函数 $\varphi(m)$ 为模 m 的既约剩余类的个数，亦即 $0, 1, \cdots, m-1$ 中与 m 互素的整数个数。设 p 是素数。当 $m = p$ 为素数时，$\{1, 2, \cdots, p-1\}$ 是模 p 的一个既约剩余系，因此 $\varphi(p) = p-1$。当 $m = p^e$ 时，集合 $\{1, 2, \cdots, p^e - 1\}$ 去掉其中的子集 $\{pi \mid 0 \leqslant i \leqslant p^{e-1} - 1\}$ 后，恰好构成模 p^e 的一个既约剩余系，因此 $\varphi(p^e) = p^e - p^{e-1} = p^{e-1}(p-1)$。对于一般的情况，设 $m = p_1^{e_1} p_2^{e_2} \cdots p_k^{e_k}$ 是 m 的完全因子分解(即 $p_1, p_2, \cdots, p_k$ 是互不相同的素数，$e_i \geqslant 1$)，如何计算 $\varphi(m)$ 呢？

首先考虑 $m=m_1m_2$，且 m_1 与 m_2 互素的情况。设 x_1 跑遍模 m_1 的一个既约剩余系，设 x_2 跑遍模 m_2 的一个既约剩余系。证明 $m_1x_2+m_2x_1$ 取遍模 m_1m_2 的一个既约剩余系。这需要证明 3 点。

首先，需要证明当(x_1,x_2)的取法不同时，$m_1x_2+m_2x_1$ 模 m_1m_2 不同余。证明它的反面。设

$$m_1x_2+m_2x_1\equiv m_1x_2'+m_2x_1'(\text{mod } m_1m_2) \tag{1.7}$$

要证明$(x_1,x_2)=(x_1',x_2')$。由式(1.7)，两边模 m_1 得

$$m_2x_1\equiv m_2x_1'(\text{mod } m_1)$$

因为 m_1 与 m_2 互素，则有 $x_1\equiv x_1'(\text{mod } m_1)$。因为 x_1,x_1'取自模 m_1 的同一个既约剩余系，则有 $x_1=x_1'$。同理 $x_2=x_2'$。

其次，证明 $m_1x_2+m_2x_1$ 与 m_1m_2 互素，即 $m_1x_2+m_2x_1$ 代表模 m_1m_2 的一个既约剩余系。若它们有大于 1 的公因子，则必有素数公因子，设为 p。由于 $p\mid m_1m_2$，m_1 与 m_2 互素，所以 $p\mid m_1$ 但 $p\nmid m_2$，或者 $p\mid m_2$ 但 $p\nmid m_1$。不妨设前者成立。由 $p\mid m_1x_2+m_2x_1$，从而有 $p\mid m_2x_1$，进而 $p\mid x_1$。因此 x_1 和 m_1 有公因子 $p>1$，这与 x_1 是模 m_1 的既约剩余系的代表矛盾。

最后，证明模 m_1m_2 的任意一个既约剩余系必有一个形如 $m_1x_2+m_2x_1$ 的代表元。设 a 与 m_1m_2 互素。由于 m_1 与 m_2 互素，可以求出(如用 1.3.1 小节中的欧氏算法)满足

$$\begin{cases} m_2x_1\equiv a(\text{mod } m_1) \\ m_1x_2\equiv a(\text{mod } m_2) \end{cases} \tag{1.8}$$

的整数 x_1,x_2。由式(1.8)可以推得

$$\begin{cases} m_1x_2+m_2x_1\equiv a(\text{mod } m_1) \\ m_1x_2+m_2x_1\equiv a(\text{mod } m_2) \end{cases}$$

由于 m_1 与 m_2 互素，则有

$$m_1x_2+m_2x_1\equiv a(\text{mod } m_1m_2)$$

同时，因为 a 与 m_i 互素，由式(1.8)可知 x_i 与 m_i 互素。

综合以上 3 点，证明了 $m_1x_2+m_2x_1$ 取遍模 m_1m_2 的一个既约剩余系。因此，$\varphi(m)=\varphi(m_1)\varphi(m_2)$。

应用这个性质，对于 $m=p_1^{e_1}p_2^{e_2}\cdots p_k^{e_k}$，有

$$\begin{aligned}\varphi(m)&=\varphi(p^{e_1})\varphi(p^{e_2})\cdots\varphi(p^{e_k})\\&=p_1^{e_1-1}p_2^{e_2-1}\cdots p_k^{e_k-1}(p_1-1)(p_2-1)\cdots(p_k-1)\end{aligned}$$

例 1.2.1　设 n 是两个不同的素数 p、q 的乘积，即 $n=pq$，则 $\varphi(n)=(p-1)(q-1)$。

1.3　典型数论算法

1.3.1　欧氏算法

设 a 和 b 是两个正整数，由 1.1.2 小节可知，存在正整数 m 和 n，使得 a 和 b 的最大公因子 $d=\gcd(a,b)$可以表示成以下形式：

$$d=am+bn \tag{1.9}$$

欧氏算法可以在 $\max\{\lg a, \lg b\}$ 的 3 次方时间内计算出 d、m 和 n。欧氏算法也用于计算同余方程

$$ax \equiv d(\bmod n)$$

的解 x。

设 $a \geqslant b$。首先用 a 除以 b 得到商 q_0 和余数 r_0，即

$$a = bq_0 + r_0 \quad 0 \leqslant r_0 < b \tag{1.10}$$

若 $r_0 = 0$，则 $b \mid a$，$d = \gcd(a,b) = b = a \cdot 0 + b \cdot 1$，已经求出了 $d = b$，$m = 0$，$n = 1$。

若 $r_0 \neq 0$，由式(1.10)不难知道 b 和 r_0 的最大公因子 $\gcd(b,r_0)$ 整除 a，因而整除 d，即 $\gcd(b,r_0) \mid \gcd(a,b)$。反过来，也容易推出 $\gcd(a,b) \mid \gcd(b,r_0)$。因此 $\gcd(a,b) = \gcd(b,r_0)$，求 a 和 b 的最大公因子转化成求 b 和 r_0 的最大公因子，而 b 和 r_0 分别小于 a 和 b。类似于式(1.10)，继续对 b 和 r_0 作带余除法，得到

$$b = r_0 q_1 + r_1 \quad 0 \leqslant r_1 < r_0 \tag{1.11}$$

若 $r_1 = 0$，则 $r_0 = \gcd(b,r_0)$。此时已求出 $d = \gcd(a,b) = r_0 = a \cdot 1 + b \cdot (-q_0)$，因而可令 $m = 1$，$n = -q_0$。

若 $r_1 \neq 0$，因为 $\gcd(b,r_0) = \gcd(r_0,r_1)$，继续上述过程，令

$$r_0 = r_1 q_2 + r_2 \quad 0 \leqslant r_2 < r_1$$

若 $r_2 = 0$，则 $d = \gcd(r_0,r_1) = r_1$。将式(1.10)代入式(1.11)，则有

$$\begin{aligned} d = r_1 &= b - q_1 r_0 = b - q_1(a - q_0 b) \\ &= a \cdot (-q_1) + b \cdot (1 + q_0 q_1) \end{aligned} \tag{1.12}$$

这样可令 $m = -q_1$，$n = 1 + q_0 q_1$。

若 $r_2 \neq 0$，可以继续上述过程，依次得到 $b > r_0 > r_1 > r_2 > \cdots$。由于这些数是非负整数，一定会有某个 $r_i = 0$。设 r_k 是第一个为 0 的 r_i。因此可以得到

$$\begin{aligned} a &= bq_0 + r_0 \\ b &= r_0 q_1 + r_1 \\ r_0 &= r_1 q_2 + r_2 \\ &\vdots \\ r_{k-3} &= r_{k-2} q_{k-1} + r_{k-1} \\ r_{k-2} &= r_{k-1} q_k \end{aligned}$$

前面已经对 $i = 0,1$ 计算出 m_i 和 n_i，使得

$$r_i = am_i + bn_i$$

对于 $i \geqslant 2$ 的情况，r_i 可以计算如下：

$$\begin{aligned} r_i = r_{i-2} - q_i r_{i-1} &= am_{i-2} + bn_{i-2} - q_i(am_{i-1} + bn_{i-1}) \\ &= a(m_{i-2} - q_i m_{i-1}) + b(n_{i-2} - q_i n_{i-1}) \end{aligned}$$

因此，得到递归公式

$$m_0 = 1, m_1 = -q_1, \cdots, m_i = m_{i-2} - q_i m_{i-1}$$
$$n_0 = -q_0, n_1 = 1 + q_0 q_1, \cdots, n_i = n_{i-2} - q_i n_{i-1}$$

由于 $r_k = 0$，因此 $d = \gcd(a, b_1) = \gcd(r_0, r_1) = \cdots = \gcd(r_{k-2}, r_{k-1}) = r_{k-1}$，而 $m = m_{k-1}$ 和 $n = n_{k-1}$ 是求得的满足式(1.9)的解。

上述迭代计算 q_i、r_i 和递归计算 m_i、n_i 的方法称为欧几里得算法(简称欧氏算法)或辗转相除法。人们通常也将欧氏算法分为欧氏算法和扩展欧氏算法,欧氏算法只计算出 d,而扩展欧氏算法可计算出 d、m 和 n。

注意,式(1.9)的整数解(m,n)并不唯一。事实上,令 t 是任意整数,将(m,n)换成$\left(m+\frac{b}{d}t,n-\frac{a}{d}t\right)$也是式(1.9)的解。欧氏算法仅求得式(1.9)的一个整数解(m,n),但由这个解不难求出其他任意解。

关于欧氏算法的计算复杂度,首先可以证明它的辗转相除的次数不大于 $2\lg a$(设$a\geqslant b$)。事实上,有

$$r_{i+2} < \frac{1}{2}r_i \tag{1.13}$$

可以如下证明式(1.13)。若 $r_{i+1}\leqslant\frac{1}{2}r_i$,则 $r_{i+2}<r_{i+1}\leqslant\frac{1}{2}r_i$。若 $r_{i+1}>\frac{1}{2}r_i$,则 r_i 除以 r_{i+1}的商为 1,即有

$$r_i = r_{i+1} + r_{i+2},\quad r_{i+2} = r_i - r_{i+1} < \frac{1}{2}r_i$$

对于两个长度不超过 $\lg a$ 比特的整数,它们作带余除法的计算复杂度不超过$(\lg a)^2$ 的某个常数倍,因此欧氏算法的计算复杂度不超过$(\lg a)^3$ 的常数倍。

系数取自某个域(如有理数域和实数域)的多项式也可按多项式次数为测度作带余除法,欧氏算法也适用于求两个多项式的最大公因式,以及适用于求多项式同余方程

$$A(x)\cdot g(x) \equiv 1(\mathrm{mod}\ f(x))$$

的解 $g(x)$(给定 $A(x)$,$f(x)$)。

例 1.3.1 设 $a=8211$,$b=2829$,用欧氏算法求其最大公因子,则有

$$\begin{aligned}8211 &= 2\times 2829 + 2553\\ 2829 &= 1\times 2553 + 276\\ 2553 &= 9\times 276 + 69\\ 276 &= 4\times 69\end{aligned}$$

因此 $\gcd(8221,2829)=69$。由上述计算可以得到

$$\begin{aligned}2553 &= 8211 - 2\times 2829\\ 276 &= 2829 - 2553 = 2829 - (8211 - 2\times 2829) = -8211 + 3\times 2829\\ 69 &= 2553 - 9\times 276 = (8211 - 2\times 2829) - 9\times(-8211 + 3\times 2829)\\ &= 10\times 8211 - 29\times 2829\end{aligned}$$

1.3.2 二次剩余判别与模 p 开平方根算法

1. 二次剩余判别算法

首先有欧拉判别条件:

$$\left(\frac{n}{p}\right) \equiv n^{\frac{p-1}{2}}(\mathrm{mod}\ p)$$

其证明分$\left(\frac{n}{p}\right)=1$和$\left(\frac{n}{p}\right)=-1$两种情况（当$\left(\frac{n}{p}\right)=0$，即$n$是$p$的倍数时，判别条件是显然成立的）。

若$\left(\frac{n}{p}\right)=1$，则存在整数$x$使得$x^2\equiv n(\mathrm{mod}\ p)$。由费马小定理（见1.2.2小节）有$x^{p-1}\equiv 1(\mathrm{mod}\ p)$，所以$n^{\frac{p-1}{2}}\equiv x^{p-1}\equiv 1(\mathrm{mod}\ p)$。

反过来考虑满足$n^{\frac{p-1}{2}}\equiv 1(\mathrm{mod}\ p)$的$n$。由于$y^{\frac{p-1}{2}}-1$是域$\mathbb{Z}_p$上的多项式，它至多有$\frac{p-1}{2}$个解（这是代数学中的一个基本结论）。上面已经证明了模p的$\frac{p-1}{2}$个二次剩余都是它的解，因此是它的全部解。换句话说，对于模p的任意一个二次非剩余m，$m^{\frac{p-1}{2}}$模p不同余1，即$p\nmid m^{\frac{p-1}{2}}-1$。另外，由费马小定理，有

$$p \mid m^{p-1}-1=(m^{\frac{p-1}{2}}-1)(m^{\frac{p-1}{2}}+1)$$

因此$p\mid m^{\frac{p-1}{2}}+1$，即$m^{\frac{p-1}{2}}\equiv -1\equiv\left(\frac{m}{p}\right)(\mathrm{mod}\ p)$。

上述欧拉判别条件给出了计算$\left(\frac{n}{p}\right)$的一个方法。这个结果一定是1或$p-1$（即$\mathbb{Z}_p$中的-1）。

不过，计算Legendre符号的更好办法是欧氏算法（见1.3.1小节）。这需要用到以下的二次互反律：设$p\neq q$是两个奇素数，则一定有

$$\left(\frac{q}{p}\right)\cdot\left(\frac{p}{q}\right)=(-1)^{\frac{p-1}{2}\cdot\frac{q-1}{2}}$$

限于篇幅，这里略去了这个定理的证明，其证明可从有关初等数论的书籍中找到。

二次互反律的意义是：设$n<p$是奇素数，则计算$\left(\frac{n}{p}\right)$的问题可轻易地转化为计算$\left(\frac{p}{n}\right)$，而后者是更小模数的问题。

一般地，设$n=q_1^{e_1}q_2^{e_2}\cdots q_k^{e_k}$是$n$的完全分解，则由欧拉判别条件可以得到

$$\left(\frac{n}{p}\right)=\left(\frac{q_1}{p}\right)^{e_1}\left(\frac{q_2}{p}\right)^{e_2}\cdots\left(\frac{q_k}{p}\right)^{e_k}$$

对于奇素数q_i，$\left(\frac{q_i}{p}\right)$的计算根据二次互反律可以转化为$\left(\frac{p(\mathrm{mod}\ q_i)}{q_i}\right)$的计算。而对于偶数的情形，则有

$$\left(\frac{2}{p}\right)=\begin{cases}1 & p\equiv 1,7(\mathrm{mod}\ 8)\\ -1 & p\equiv 3,5(\mathrm{mod}\ 8)\end{cases}$$

另外，对$\left(\frac{u}{p}\right)$，还有$\left(\frac{u}{p}\right)=\left(\frac{-1}{p}\right)\left(\frac{p-u}{p}\right)$。因此，如果$\frac{p-1}{2}<u<p$，可以通过计算$\left(\frac{p-u}{p}\right)$和$\left(\frac{-1}{p}\right)$来计算$\left(\frac{u}{p}\right)$，其中

$$\left(\frac{-1}{p}\right)=\begin{cases}1 & p\equiv 1(\mathrm{mod}\ 4)\\ -1 & p\equiv 3(\mathrm{mod}\ 4)\end{cases}$$

上述计算方法中要用到 n 的因子分解，而因子分解是一个比模指数运算要难得多的问题。为了弥补这个缺陷，引进 Jacobi 符号：设 m 为正奇数，$m=p_1p_2\cdots p_k$，p_i 为奇素数，并可以重复出现。对于与 m 互素的奇整数 n，定义 Jacobi 符号 $\left(\frac{n}{m}\right)$ 如下：

$$\left(\frac{n}{m}\right)=\left(\frac{n}{p_1}\right)\left(\frac{n}{p_2}\right)\cdots\left(\frac{n}{p_k}\right)$$

在 Jacobi 符号的定义中要求 n 与 m 互素，且 m 为奇数。对于 Jacobi 符号，可以证明同样有

$$\left(\frac{2}{m}\right)=\begin{cases}1 & p\equiv 1,7(\bmod 8)\\ -1 & p\equiv 3,5(\bmod 8)\end{cases},\quad \left(\frac{-1}{m}\right)=\begin{cases}1 & p\equiv 1(\bmod 4)\\ -1 & p\equiv 3(\bmod 4)\end{cases}$$

更重要的是，可以从 Legendre 符号的二次互反律推出以下关于 Jacobi 符号的二次互反律。设 m 和 n 为互素的两个正奇数，则

$$\left(\frac{n}{m}\right)\cdot\left(\frac{m}{n}\right)=(-1)^{\frac{m-1}{2}\cdot\frac{n-1}{2}}$$

因此，当 $n<m$ 为奇数时，为计算 $\left(\frac{n}{m}\right)$，只需计算 $\left(\frac{m}{n}\right)$。作带余除法

$$m=nq+r,\quad 0<r<n$$

则 $\left(\frac{m}{n}\right)=\left(\frac{r}{n}\right)$。设 $r=2^e r_0$，其中 r_0 为奇数，则

$$\left(\frac{m}{n}\right)=\left(\frac{2}{n}\right)^e\left(\frac{r_0}{n}\right)$$

于是问题变成 $\left(\frac{r_0}{n}\right)$ 的计算，而 r_0 和 n 是分别小于 n 和 m 的互素的奇数。这样，多次"辗转"运用带余除法和二次互反律，就可以计算出 $\left(\frac{n}{m}\right)$，而不涉及整数的因子分解。

该计算方法的效率至少不低于欧氏算法，因为余数提出 2 的方幂后会变得更小。另外，因为 Jacobi 符号的平方总是等于 1，因此还可以将余数去掉它的平方因子，即若 $r=w^2r'$，则

$$\left(\frac{r}{n}\right)=\left(\frac{r'}{n}\right)$$

2. 模 p 开平方根算法

当 $\left(\frac{a}{p}\right)=1$ 时，$x^2\equiv a(\bmod p)$ 有解，求这个方程的解的问题称为求模 p 平方根问题。下面给出一个求模 p 的平方根算法。

首先给出元素的阶的概念。设 a 是 Z_p^* 中的一个元素，称使得 $a^t=1$ 的最小正整数 t 为 a 的阶。可以证明 $\mathbb{Z}_p$ 中存在阶为 $p-1$ 的元素，记为 g，称其为模 p 的一个原根。$\mathbb{Z}_p^*$ 中的任意一个元素都可以唯一地表示成 g 的方幂，即存在 $0\leqslant i<p$ 使得 $a\equiv g^i(\bmod p)$。因此有 $\left(\frac{a}{p}\right)=\left(\frac{g}{p}\right)^i$，进而 $\left(\frac{g}{p}\right)=-1$（即 g 是模 p 的二次非剩余），并且 $\left(\frac{a}{p}\right)=1$ 当且仅当 i 是偶数。

设 $p-1=2^s t$，t 为奇数，$s\geqslant 1$。因为 2^s 与 t 互素，所以存在整数 u,v 使得 $tu+2^s v=1$，进而有

$$g\equiv g^{tu}\cdot g^{2^s v}(\mathrm{mod}\ p),\quad g^i\equiv b^i\cdot g^{2^s vi}(\mathrm{mod}\ p)$$

其中 $b=g^{tu}$。因为 tu 是奇数，$b=g^{tu}$ 是模 p 的二次非剩余，即 $\left(\frac{b}{p}\right)=\left(\frac{g}{p}\right)^{tu}=-1$。由于

$$b^{2^s}\equiv g^{2^s tu}\equiv 1(\mathrm{mod}\ p),\quad (g^{2^s v})^t\equiv g^{2^s tv}\equiv 1(\mathrm{mod}\ p)$$

并且对 $k<2^s$ 和 $j<t$，b^k 和 $(g^{2^s v})^j$ 模 p 都不同余 1，因此 $b(\mathrm{mod}\ p)$ 和 $g^{2^s v}(\mathrm{mod}\ p)$ 的阶分别为 2^s 和 t。令 $e\equiv i(\mathrm{mod}\ 2^s)$，$c\equiv g^{2^s vi}(\mathrm{mod}\ p)$，则

$$a\equiv b^e\cdot c(\mathrm{mod}\ p)\tag{1.14}$$

此处 e 是偶数，因为 i 是偶数。

给定 a，然而并不知道 i 和 e 的具体值。设 e 的二进制展开式为

$$e=2e_1+2^2e_2+\cdots+2^{s-1}e_{s-1},\quad e_i\in\{0,1\}\tag{1.15}$$

下面的算法将依次决定出 $e_1,e_2,\cdots,e_{s-1}$，即求出 e，进而由式(1.14)求出 $c(\mathrm{mod}\ p)$。注意到，$b^{\frac{e}{2}}$ 是 $b^e(\mathrm{mod}\ p)$ 的一个平方根，而 $c(\mathrm{mod}\ p)$ 的一个平方根是 $c^{\frac{t+1}{2}}(\mathrm{mod}\ p)$，这是因为

$$(c^{\frac{t+1}{2}})^2\equiv c^{t+1}\equiv c(\mathrm{mod}\ p)$$

于是得到 a 的平方根 $\pm b^{\frac{e}{2}}\cdot c^{\frac{t+1}{2}}(\mathrm{mod}\ p)$。

求 e_i 的方法如下。因为

$$(2e_1+2^2e_2+\cdots+2^{s-1}e_{s-1})2^{s-2}t\equiv 2^{s-1}e_1(\mathrm{mod}\ 2^s)$$

对式(1.14)两边都进行 $2^{s-2}t$ 次方得到

$$a^{2^{s-2}t}\equiv(b^{2^{s-1}})^{e_1}(\mathrm{mod}\ p)$$

因为 $b^{2^s}\equiv 1(\mathrm{mod}\ p)$，$b^{2^{s-1}}$ 模 p 不同余 1，故有 $b^{2^{s-1}}\equiv -1(\mathrm{mod}\ p)$。这样得到

$$a^{2^{s-2}t}\equiv(-1)^{e_1}(\mathrm{mod}\ p)\tag{1.16}$$

通过计算 $a^{2^{s-2}t}$，从而决定出 e_1。求出 e_1 后可如下求 e_2。因为

$$a\cdot b^{-2e_1}\equiv b^{2^2e_2+\cdots+2^{s-1}e_{s-1}}\cdot c(\mathrm{mod}\ p)$$

用求式(1.16)的办法，有

$$(a\cdot b^{-2e_1})^{2^{s-3}t}\equiv(-1)^{e_2}(\mathrm{mod}\ p)$$

从而同样地决定出 e_2。以此类推，可以类似地得到 e_3、e_4、$\cdots$、e_{s-1}。

可以证明，上述算法中的 b 可以换成模 p 的任意一个非二次剩余。这样，随机地选取 1 到 $p-1$ 中的整数，有一半的概率可以作为算法所需要的 b。

算法 1.3.1 求模 p 的平方根

输入：素数 p，整数 a，$\left(\frac{a}{p}\right)=1$

输出：整数 x，使得 $x^2\equiv a(\mathrm{mod}\ p)$

1. 选取模 p 的一个二次非剩余 b；
2. 分解 $p-1=2^s t$，t 是奇数，$s\geqslant 1$；

3. $e \leftarrow 0$；

4. 若 $s>1$，则对 $i=1\sim s-1$，若 $(a\cdot b^{-e})^{2^{s-(i+1)}t}\equiv -1 \pmod p$，则令 $e\leftarrow e+2^i$；

5. $h\leftarrow ab^{-e} \pmod p$，$x\leftarrow \pm b^{\frac{e}{2}}h^{\frac{t+1}{2}} \pmod p$；

6. 返回 x。

例 1.3.2　当 $p\equiv 3 \pmod 4$ 时，$s=1$，$t=\dfrac{p-1}{2}$，可取 $b=p-1$，则 a 模 p 的平方根是 $x\equiv \pm a^{\frac{p+1}{4}} \pmod p$。

例 1.3.3　当 $p\equiv 5 \pmod 8$ 时，$s=2$，$t=\dfrac{p-1}{4}$，可取 $b=2^{-1} \pmod p$，则

(1) 当 $a^{\frac{p-1}{4}}\equiv -1 \pmod p$ 时，a 模 p 的平方根是 $x\equiv \pm 2^{\frac{p-1}{4}}a^{\frac{p+3}{8}} \pmod p$；

(2) 否则，a 模 p 的平方根是 $x\equiv \pm a^{\frac{p+3}{8}} \pmod p$。

1.3.3　素数检测算法

给定一个正奇数 n，判断它是否为素数，称为素数检测。下面介绍一种比较常用的概率性素数检测方法。这种方法有着较高的运算效率。

当 n 是素数时，由费马小定理，对于 $(0,n)$ 区间的任意整数 a，都有

$$a^{n-1}\equiv 1 \pmod n \tag{1.17}$$

因此，若有整数 $a\in(0,n)$ 使得式(1.17)不成立，则 n 一定不是素数。若 a 使得式(1.17)成立，则不能判定 n 是否为素数；但如果能证明有这样一个性质："当 n 不是素数时，$(0,n)$ 中的整数使得式(1.17)成立的 a 的比率 $\rho<\dfrac{1}{2}$"，则可以通过独立地选取 k 个不同的 a，逐个判别式(1.17)是否成立。若有一个 a 使得式(1.17)不成立，则已经判断出 n 不是素数，算法输出检测结果并终止；若这 k 个不同的 a 都满足式(1.17)，由于此时 n 不是素数的概率不大于 ρ^k，当 k 较大时 ρ^k 很小，这时就能以很大的正确概率 $1-\rho^k$ 判断出 n 是素数。换句话说，"n 不是素数却被误判为是素数"的概率不大于 ρ^k。这样，就得到了一种概率性的素数判定方法。这种方法是"证伪"的，即返回的否定性结论一定正确。

但遗憾的是，不存在上述小于 $\dfrac{1}{2}$ 的 ρ。有一类非素数，称为 Carmichael 数，它们对任意整数 $a\in(0,n)$，式(1.17)都不成立。例如，$561=3\times 11\times 17$ 就是一个 Carmichael 数。

下面讲述的 Miller-Rabin 检测方法克服了上述缺陷。继续考虑式(1.17)，将两边开平方，对于素数 n，由于 1 模 n 的平方根必然是 $\pm 1 \pmod n$，则有

$$a^{\frac{n-1}{2}}\equiv 1 \pmod n \quad 或 \quad a^{\frac{n-1}{2}}\equiv -1 \pmod n$$

若前者成立，就可以继续开平方得到 $a^{\frac{n-1}{4}}\equiv \pm 1 \pmod n$。因此，设 $n-1=2^s t$，$s\geqslant 1$，t 是奇数，若 n 是素数，则

$$a^{\frac{n-1}{2}}, a^{\frac{n-1}{4}}, \cdots, a^{\frac{n-1}{2^s}} \tag{1.18}$$

中第一个模 n 不同余 1 的必同余 -1。由于 $\frac{n-1}{2^s}=t$，式(1.18)中的数从右向左依次是 a^t, a^t 的平方，再平方，……。若 a^t 模 n 不同余 1，则 $a^t \equiv -1 \pmod n$。

算法 1.3.2 Miller-Rabin 素数检测算法

输入：正奇数 n，$n-1=2^s t$，$s \geqslant 1$，t 为奇数，检测轮数 k

输出：整数"n 是素数"或"n 不是素数"

1 重复以下步骤 k 轮；

  1.1 独立随机地选取整数 $a \in (0,n)$；

  1.2 计算 $b \equiv a^t \pmod n$；

  1.3 若 $b=1$ 或者 $n-1$，结束本轮进入下一轮，否则依次计算 $b^2 \pmod n$、$b^4 \pmod n$、…、$b^{2^{s-1}} \pmod n$ 若这些数出现 $n-1$，结束本轮进入下一轮，否则输出"n 不是素数"并终止算法；

2 输出"n 是素数"。

当 Miller-Rabin 素数检测算法输出"n 不是素数"时，算法对某个 a 发现式(1.18)中每个数模 n 既不是 1 也不是 -1，由上述分析知，n 一定不是素数(即该算法"证伪")。

可以从数学上严格证明，Miller-Rabin 素数检测算法的一轮测试将非素数判别为素数的误判概率小于 $\frac{1}{4}$，即对任意非素数的奇数 n，区间 $(0,n)$ 中至多有 $\frac{1}{4}$ 的整数 a 能通过"Miller-Rabin 素数检测算法"中的 1.3 步的测试。取 $k=20$，则 Miller-Rabin 素数检测算法将非素数 n 判断为素数的概率小于 2^{-40}(小于一万亿分之一)。

1.3.4 因子分解算法

设 n 是需要分解因子的正奇数，如果能够找到两个整数 a 和 b，使得

$$a^2 \equiv b^2 \pmod n \tag{1.19}$$

则 $n \mid (a+b)(a-b)$。当 $n \nmid (a+b)$ 或 $n \nmid (a-b)$ 时，$\gcd(n, a+b)$ 或 $\gcd(n, a-b)$ 是 n 的真因子，从而将 n 分解成两个小一些的整数的乘积。当 $n \mid a \pm b$ 时，即 $a \equiv \pm b \pmod n$，式(1.19)对 n 的分解无帮助，应寻找新的 a 和 b。

为获得满足式(1.19)的 a 和 b，选取 m 个 a_i，计算 $c_i \equiv a_i^2 \pmod n$，将其中某些 c_i 相乘，希望得到一个平方数。为此，挑选那些能完全分解为小素因子(可重复出现)的乘积的 c_i，即设 $p_1, p_2, \cdots, p_k$ 为不同的小素数，

$$a_i^2 \equiv p_1^{e_{i1}} p_2^{e_{i2}} \cdots p_k^{e_{ik}} \pmod n \tag{1.20}$$

若干个式(1.20)的右边相乘为平方数等价于若干个指数向量$(\boldsymbol{e}_{i1}, \boldsymbol{e}_{i2}, \cdots, \boldsymbol{e}_{ik})$的和向量的每个分量都是偶数。显然，这是考虑相应分量之和模 2 为 0 的问题。因此，只需将 $\boldsymbol{e}_{ij}$ 模 2 变成 0 或 1(仍然记为 $\boldsymbol{e}_{ij}$)，然后求解 0～1 向量$(\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_m)$，使得

$$\boldsymbol{x}_1(e_{11}, e_{12}, \cdots, e_{1k}) + \boldsymbol{x}_2(e_{21}, e_{22}, \cdots, e_{2k}) + \cdots + \boldsymbol{x}_m(e_{m1}, e_{m2}, \cdots, e_{mk}) = 0$$

即

$$\begin{pmatrix} e_{11} & e_{21} & \cdots & e_{m1} \\ e_{12} & e_{22} & \cdots & e_{m2} \\ \vdots & \vdots & \ddots & \vdots \\ e_{1k} & e_{2k} & \cdots & e_{mk} \end{pmatrix} \begin{pmatrix} \boldsymbol{x}_1 \\ \boldsymbol{x}_2 \\ \vdots \\ \boldsymbol{x}_m \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \tag{1.21}$$

因此,以上因子分解方法可以分成两个步骤:

(1) 收集足够多的形如式(1.20)的关系式;

(2) 对收集的指数向量,求解 0～1 向量($\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_m$)满足线性方程组(1.21)。

对于很大的整数 n,实际需要的 k 和 m 可能很大,这样不仅需要收集很多的关系式,而且还面临着一个大规模线性方程组的求解问题。

1. 连分式因子分解法

连分式因子分解方法利用了$\sqrt{n}$的连分式展开式的各个渐近分数$\frac{r_i}{q_i}$一定满足以下形式的方程

$$q_i^2 \equiv (-1)^{i-1}Q_i(\bmod n) \quad 0 < Q_i < 2\sqrt{n}$$

由于 Q_i 相对 n 较小,它的素因子分解有可能落入预先选定的 k 个素数 p_1, $p_2,\cdots,p_k$ 之内(可以依次用 $p_1,p_2,\cdots,p_k$ 这 k 个素数去除 Q_i 判断是否如此)。$(-1)^{i-1}$的指数 $i-1$ 也可以当作一个向量分量来处理。

关于连分式因子分解方法的详细论述参见文献[6]。

2. 二次筛法

设$\lfloor\sqrt{n}\rfloor$表示不大于$\sqrt{n}$的最大整数,取小整数 x,令

$$f(x) = (x + \lfloor\sqrt{n}\rfloor)^2 - n$$

则 $0<f(x)<2\sqrt{n}x+x^2$。当 $x<n^{\frac{1}{4}}$时,$f(x)$近似等于 $2\sqrt{n}x$,它的素因子可能落入 $p_1,p_2,\cdots,p_k$ 之内。

显然,可以使用其他方式定义的二次多项式代替 $f(x)$。

关于二次筛法的详细论述参见文献[6]。

1.4　应用举例

1.4.1　RSA 密码算法

在 RSA 密码算法中,一个用户选取两个不同的大素数 p 和 q,令 $n=pq$,则 n 的欧拉函数 $\varphi(n)=(p-1)(q-1)$。用户选取与 $\varphi(n)$互素的整数 $0<e<\varphi(n)$,利用欧氏算法,计算出一个整数 d,满足

$$ed \equiv 1(\bmod \varphi(n)) \tag{1.22}$$

用户将 n 和 e 作为公开密钥(公钥)向外发布,d 作为秘密密钥(私钥)秘密保存。

显然,如果 p 和 q 被泄露,则知道 p 和 q 的任何人都知道了 $\varphi(n)$,从而可以计算出 d。因此 p 和 q 也必须保密。

RSA 加密过程：将要加密的明文消息编码成整数 m，$0<m<n$。密文 c 定义为

$$c \equiv m^e \pmod n, \quad 0<c<n \tag{1.23}$$

RSA 解密过程：公、私钥对(e,d)的拥有者计算

$$m' = c^d \pmod n \tag{1.24}$$

解密的正确性：由式(1.23)和式(1.24)有

$$m' \equiv m^{ed} \pmod n$$

由式(1.22)，存在整数 k 使得 $ed=1+k\varphi(n)$。由欧拉定理有

$$m^{ed} \equiv m \cdot (m^{\varphi(n)})^k \equiv m \cdot 1^k \equiv m \pmod n$$

所以

$$m' \equiv m \pmod n$$

由于 m'和 m 都是落在区间$(0,n)$的整数，因此 $m'=m$。

可以应用中国剩余定理来加速 RSA 的解密计算。RSA 解密方知道 d、p 和 q，令

$$d_1 \equiv d \pmod{p-1}, \quad 0<d_1<p-1$$
$$d_2 \equiv d \pmod{q-1}, \quad 0<d_2<q-1$$

对收到的密文 c，令

$$c_1 \equiv c \pmod p, \quad 0<c_1<p$$
$$c_2 \equiv c \pmod q, \quad 0<c_2<q$$

再令

$$\begin{aligned} m_1 &\equiv m \pmod p, \quad 0<m_1<p \\ m_2 &\equiv m \pmod q, \quad 0<m_2<q \end{aligned} \tag{1.25}$$

由于 $m\equiv c^d \pmod n$，对此式两边模 p，再根据费马小定理有

$$m_1 \equiv c_1^d \equiv c_1^{d_1} \pmod p$$

同理可以得到

$$m_2 \equiv c_2^{d_2} \pmod q$$

计算出 m_1 和 m_2 后，利用中国剩余定理(见 1.2.1 小节)，由式(1.25)即可以计算出 m。

这种计算方法的优势是做计算量大的模指数运算时，使用的模数 p 和 q 比直接计算时的模数 n 要小。另外，模指数运算的指数也变小了。相比这个优势，额外增加的中国剩余定理的计算量显得微不足道。

1.4.2 Rabin 密码算法

Rabin 密码算法与 RSA 密码算法采用同样的模数。一个用户拥有两个不同的素数 p、q 作为私钥，将它们的乘积 $n=pq$ 作为公钥公开。在 Rabin 密码算法中，加密不再是模 n 的高次幂，而仅仅是模 n 平方运算，即消息(编码为满足 $0<m<n$ 的整数 m)对应的密文是

$$c \equiv m^2 \pmod n \tag{1.26}$$

解密者由于拥有素数 p、q，他设法先求出 $m_1\equiv m \pmod p$ 和 $m_2\equiv m \pmod q$，再

利用中国剩余定理求解。设 $c_1 \equiv c(\text{mod } p)$ 和 $c_2 \equiv c(\text{mod } q)$，由式(1.26)，有

$$c_1 \equiv m_1^2(\text{mod } p) \tag{1.27}$$

$$c_2 \equiv m_2^2(\text{mod } q) \tag{1.28}$$

利用算法 1.3.1 中的模素数开平方算法，可求得式(1.27)的两个解(它们互为模 p 相反数)和式(1.28)的两个解(它们互为模 q 相反数)。这样共有 4 组解($\pm m_1(\text{mod } p)$，$\pm m_2(\text{mod } q)$)。按照中国剩余定理，每一组解可导出一个 $m(\text{mod } n)$。因此，解密结果有 4 个，不唯一。克服这个缺点的一个方法是，在实际应用中，在加密时先对明文进行冗余信息添加(如在明文后面复制原始明文消息的后若干个比特)。这样，一个合法密文的 4 个模 n 平方根很可能只有一个具有这种冗余性质，解密者就接受这个根为明文。

在弱的安全意义下(只考虑被动攻击敌手)，Rabin 密码算法的安全性可证明与因子分解问题等价。而 RSA 密码算法尚不能证明这一点。

1.5 注记

初等数论是很多抽象代数概念如群、环、域、模的源泉，也是通信、信息安全等领域中广泛应用的工具。本章简要介绍了初等数论中的基本概念、基本原理和一些数论算法，更多的内容请参看本章的参考文献。其中文献[1]～[4]是关于初等数论的中文版大学和研究生教材，文献[5]是英文版研究生教材。有关数论算法，请参阅文献[4]、[6]、[7]，文献[6]、[7]是两本极好的教科书。数论中有很多悬而未决的问题，请参阅文献[8]、[9]，其中文献[9]专门论述素数问题，有些问题就是针对信息安全需求提出来的。

关于素数检测，本章只给出了一种概率性素数检测方法，虽然效率高，但概率性素数检测方法的缺陷是，当它判别一个奇数为素数时，它不能断定该数百分之百的是素数。为弥补这个没有“完全证实”的缺陷，确定性素数检测(也叫确定性素性测试或素性证明)能够明确地、准确无误地说明一个整数是否为素数。目前有 3 种方法可以实现确定性素性判定，分别是 Jacobi 和素性证明、椭圆曲线素性证明和 AKS 素性判定。

1. Jacobi 和素性证明

作为确定性素性判定方法之一，Jacobi 和素性证明判定整数 n 为素数的时间复杂度是 $O((\ln n)^{c\ln\ln\ln n})$，其中 c 是某一常数。特别地，当 $n<2^{512}$ 时，$\ln\ln\ln n<1.78$，因此接近于一个多项式时间算法。对几百位十进制大的整数，使用 Jacobi 和素性证明判定法及现有的普通计算机能够在数分钟内完成计算。因而，Jacobi 和素性证明是实用的。

Jacobi 和素性证明在实际应用时是一个随机算法，对于每个 $k\geqslant 1$，算法至少以 $1-2^{-k}$ 的概率在 $O(k(\ln n)^{c\ln\ln\ln n})$ 步内终止。当算法终止时，算法总能给出一个整数是否为素数的正确结果。

Jacobi 和素性证明的缺点是不能提供一个可以在比算法运行短很多的时间内验证运算结果正确性的数据(“素性证书”)。另一个缺点是 Jacobi 和素性证明不像 Miller-Rabin 概率性素数测试那样容易编程，它的计算机编程很复杂，返回结果的代

码也不简洁。

2. 椭圆曲线素性证明

椭圆曲线素性证明是另一种确定性素性判定方法，通常也称为 Atkin 测试。对任何$u>0$，在启发式假设下，该算法的平均运行时间是 $O((\ln n)^{6+u})$ 比特操作。Atkin 测试现在能够证明超过 1000 位的十进制长的奇数是否为素数。

Atkin 测试胜过 Jacobi 和素性证明的一个优点是，Atkin 测试能够产生一个简短的素性证书，并且利用素性证书能够有效地验证整数的素性。但 Jacobi 和素性证明与 Atkin 测试的算法细节都相当复杂。

3. AKS 素性判定

AKS 素性判定是 2002 年 Agrawal、Kayal 和 Saxena 发现的一种素性判定方法。AKS 素性判定的算法原理比较简单，并且是一个完全多项式时间的确定性素性判定方法。

关于因子分解，除了正文中提到的方法外，比较有名的还有数域筛法和椭圆曲线因子分解方法。

数域筛法是目前最有效的大数因子分解算法。其思想与二次筛法相同，但不是使用整数的同余关系式，而是使用代数数域中的代数整数的同余式。

椭圆曲线因子分解是一种与上述方法原理不同的因子分解法。它能够用来发现大整数的较小的素因子。使用椭圆曲线因子分解法找出整数 n 的素因子 p 的平均运行时间是$O(\exp[(2^{1/2}+o(1))(\ln p)^{1/2}(\ln\ln p)^{1/2}])$。

椭圆曲线因子分解方法的运行时间依赖于整数 n 的最大素因子的规模，它是一种特殊目的的因子分解方法，倾向性地被用于分解出整数的小素因子。目前它能够找到大整数的 40 位十进制以内的素因子。

找到整数的小素因子后，利用其他因子分解方法能够继续寻找整数的其他素因子。

参 考 文 献

[1] 华罗庚. 数论导引. 北京：科学出版社，1957
[2] 闵嗣鹤，严士键. 初等数论(第三版). 北京：高等教育出版社，2003
[3] 潘承洞，潘承彪. 初等数论(第二版). 北京：北京大学出版社，2003
[4] 裴定一，祝跃飞. 算法数论. 北京：科学出版社，2002
[5] Ireland K, Rosen M. A Classical Introduction to Modern Number Theory. New York: Springer-Verlag, 1990 and Beijing: World Publishing Corporation
[6] Cohen H. A Course in Computational Algebraic Number Theory. Berlin: Springer-Verlag, 1993 and Beijing: World Publishing Corporation, 1997
[7] Gathen J, Gerhard J. Modern Computer Algebra. Cambridge: Cambridge University Press, 1999 and Beijing: World Publishing Corporation, 2001
[8] R. K. 盖伊著. 张明尧译. 数论中未解决的问题. 第二版. 北京：科学出版社，2003
[9] P. 里本伯姆著. 孙淑玲，冯克勤译. 博大精深的素数. 北京：科学出版社，2007

第 2 章　代数方法与技术

代数学的主要研究对象是各种各样的代数结构，即具有一些代数运算的集合。代数学在信息安全中有着广泛的应用。本章主要介绍代数学中的一些方法和技术。2.1 节至 2.5 节介绍了群、环、域、模和格的一些基本概念和基本知识，2.6 节介绍了有限域和 Galois 环，而在 2.7 节介绍了代数学中的一些典型算法，以及代数学在密码学中的一些应用。

2.1　群

群是代数学中的基本概念，它是一种只含有单个运算的代数结构，它的运算法则与数的运算法则类似，在自然科学的许多领域都有着广泛的应用。在这一节，主要介绍群的一些基本概念和基本性质。

2.1.1　定义及基本性质

设 S 是一非空集合，我们把 $S\times S\to S$ 的一个映射$\circ$称为 S 上的(二元)运算。对于 S 上的一个二元运算$\circ$，为了方便起见，也把 $a,b\in S$ 的像$\circ(a,b)$记做 $a\circ b$，或者省略$\circ$，只简单地写做 ab。二元运算是我们非常熟悉的一类运算，比如说整数的加法和乘法运算就是定义在整数集合上的二元运算。这节讨论只有一种代数运算的代数结构，这种代数结构称为群，请看下面的定义。

定义 2.1.1　我们说一个非空集合 G 对于 G 上的一个二元运算$\circ$来说作成一个群，如果满足：

1) $\circ$是结合的，即对任何 $a,b,c\in G$，$a\circ(b\circ c)=(a\circ b)\circ c$。

2) G 中存在一个元素 e 满足：$\forall a\in G$，$a\circ e=e\circ a=a$。这个元素称为 G 中的单位元。有时也把单位元 e 写成 1_G 或 1。

3) 对 $\forall a\in G$，存在一个元素 $a^{-1}\in G$ 满足 $a\circ a^{-1}=a^{-1}\circ a=e$，这个元素称为 a 的逆元。

4) 进一步，如果 G 中的元素还满足 $\forall a,b\in G$，$a\circ b=b\circ a$，则 G 称为交换群或 Abel 群。这时也把$\circ$表示成$+$，同时把单位元 e 写成 0。因此，交换群也称为加法群。

实际上，群对我们来说并不陌生，请看下面的例子。

例 2.1.1　所有整数的集合 Z 在加法运算下构成一个交换群$(Z,+)$。在这个群中，二元运算就是通常的加法运算，单位元是 0，这是因为 0 加任何数都还等于原来的数。而一个整数 a 的逆元是$-a$。又因为整数的加法满足交换律，即对任意的整数 a 和 b，都有 $a+b=b+a$，因此$(Z,+)$是一个交换群。

例 2.1.2　只含有一个元素 e 的集合在运算 $e\circ e=e$ 下构成一个群。在这个群

中，单位元就是 e，而 e 的逆元还是 e。

例 2.1.3 令 $G=\{0,1,2,\cdots,5\}$，则 G 在运算"$a\circ b=a+b$ 除 6 后的余数"下构成一个群。在这个群中，单位元是 0，1 和 5 互为逆元，2 和 4 互为逆元，3 的逆元是其本身。

例 2.1.4 元素在数域 K 中的全体 n 阶可逆矩阵对于矩阵的乘法构成一个群，这个群记为 $GL_n(K)$，称为 n 级一般线性群；$GL_n(K)$ 中全体行列式为 1 的矩阵对于矩阵乘法也构成一个群，这个群记为 $SL_n(K)$，称为特殊线性群。显然，一般线性群和特殊线性群都不是交换群。

例 2.1.5 集合 $\{1,2,\cdots,n\}$ 上的所有置换在置换的复合运算下构成一个非交换群，这个群称为对称群。

从上面群的例子可以看出，一个群既可只含有有限个元素，也可含有无限个元素。请看下面的定义。

定义 2.1.2 一个群 G 称为有限群(无限群)，如果 G 中含有有限多个元素(无限多个元素)。群中元素的个数称为 G 的阶，用 $|G|$ 表示。若 G 为无限群，记 $|G|=\infty$。

容易证明在一个群中，单位元和元素的逆元是唯一的，而且对 $\forall a,b\in G$，$(a\circ b)^{-1}=b^{-1}\circ a^{-1}$，我们约定：

$$a^0=e$$

$$a^n=\underbrace{a\circ a\circ\cdots\circ a}_{(n\text{个})}\quad 或\quad na=\underbrace{a+a+\cdots+a}_{(n\text{个})}$$

$$a^{-n}=(a^{-1})^n$$

显然有：$a^na^m=a^{m+n}$，$(a^n)^m=a^{mn}$。

定义 2.1.3 设 G 是群，a 是 G 中的一个元素。如果存在正整数 m，使得 $a^m=1$，则称 a 是有限阶的元素，而把满足 $a^m=1$ 的最小的正整数 m 叫做元素 a 的阶，用 $o(a)$ 或 $|a|$ 表示。否则称 a 是无限阶的元素。

定理 2.1.1 设 a 是群 G 中的一个有限阶元素，$o(a)=m$，则对任意的正整数 n，$a^n=1$ 当且仅当 $m\mid n$。

证明：充分性：假设 $m\mid n$，则存在 t 使 $n=mt$，所以 $a^n=a^{mt}=(a^m)^t=1^t=1$。

必要性：假设 $a^n=1$，$n=qm+r$，其中 q 和 r 都是非负整数，$0\leqslant r<m$。那么

$$1=a^n=a^{qm+r}=a^{qm}a^r=(a^m)^qa^r=1\cdot a^r=a^r$$

但由于 $0\leqslant r<m$，根据元素阶的定义，m 是使 $a^m=1$ 成立的最小正整数，因此 $r=0$，所以 $n=qm$，即 $m\mid n$。

定理 2.1.2 设 a 是群 G 中的一个有限阶元素，$o(a)=m$。则对任意的正整数 k，a^k 的阶为 $\frac{m}{(k,m)}$，其中 (k,m) 表示 k 和 m 的最大公因子。

证明：假设 $d=(k,m)$，$o(a^k)=n$，则 $\left(\frac{m}{d},\frac{k}{d}\right)=1$，$a^{kn}=(a^k)^n=1$。根据定理 2.1.1，$m\mid kn$，所以 $\frac{m}{d}\mid\left(\frac{k}{d}\cdot n\right)$，从而 $\frac{m}{d}\mid n$。但显然 $(a^k)^{\frac{m}{d}}=1$，再次根据定理 2.1.1 知 $n\,\Big|\,\frac{m}{d}$。因此，$n=\frac{m}{d}$。

定义 2.1.4　一个群 G 称为循环群，如果存在一个元素 $a \in G$ 使得 $G=\langle a\rangle$。这样的元素 a 称为 G 的生成元。

显然，任何的循环群都是交换群。例 2.1.1 是一个循环群，生成元为 1。Z_n 也是一个循环群，且 $Z_n=\langle[1]\rangle$，而且对任何一个与 n 互素的整数 t，$[t]$都是 Z_n 的生成元。

定理 2.1.3　任意循环群的子群仍是循环群。

证明：设 $G=\langle a\rangle$是一循环群，H 是 G 的一个子群。不妨设 $H\neq\{1\}$。因为 $a^n\in H\Rightarrow a^{-n}\in H$，所以 a 的某一正次幂一定在 H 中。设 d 是使得 $a^d\in H$ 的最小正整数，即 $d=\min\{n\in \mathbf{Z} \mid n>0$ 且 $a^n\in H\}$。下面证 $H=\langle a^d\rangle$。任给 $h\in H$，存在 $s\in\mathbf{Z}$ 使 $h=a^s$，写 $s=qd+r, 0\leqslant r<d$，则有 $a^s=a^{qd}\cdot a^r\in H$，从而 $a^r\in H$。根据 d 的选取，可知 $r=0$，所以 $H=\langle a^d\rangle$.

定理 2.1.4　设 $G=\langle a\rangle$是一有限阶的循环群，阶为 m。那么

1）如果 d 是 m 的一个因子，则 G 包含且只包含一个指数为 d 的子群。而且对 m 的任一因子 f，G 正好包含一个阶为 f 的子群。

2）设 f 是 m 的因子，则 G 中有 $\phi(f)$个阶为 f 的元素。其中 $\phi(f)$是 Euler 函数，即小于 f 且与 f 互素的正整数的个数。

3）G 正好有 $\phi(m)$个生成元，且每一生成元都具有形式 a^r，其中$(r,m)=1$。

证明：

1）假设 d 已给定，则根据定理 2.1.2，$\langle a^d\rangle$是阶为$\frac{m}{d}$的子群，因此指数为 d。假设$\langle a^k\rangle$是另一个指数为 d 的子群，则根据定理 2.1.2，$|\langle a^k\rangle|=\frac{m}{d}$。但另一方面，$|\langle a^k\rangle|=\frac{m}{(m,k)}$。所以 $d=(m,k)$，因此有 $d\mid k$，$a^k\in\langle a^d\rangle$，$\langle a^k\rangle\subset\langle a^d\rangle$。但由于$\langle a^k\rangle$和$\langle a^d\rangle$有相同的阶，所以$\langle a^k\rangle=\langle a^d\rangle$。

注意到，对 m 的任一因子 f，阶为 f 的子群正好(一定)是指数为$\frac{m}{f}$的子群，容易证明 G 正好包含一个阶为 f 的子群。

2）设 a^k 是 G 中的一个元素，则 a^k 的阶是$\frac{m}{(k,m)}$。所以 a^k 的阶是 $f\Leftrightarrow m=f\cdot(k,m)\Leftrightarrow(k,m)=\frac{m}{f}$。令 $k=h\cdot\frac{m}{f}$，则$(k,m)=\frac{m}{f}\Leftrightarrow(h,f)=1$。而 h 的个数正好是 $\phi(f)$。

3）因为 G 的生成元一定是阶为 m 的元素，根据 2)，所以 G 的生成元的个数为 $\phi(m)$。设 $a^r\in G$ 是 G 的一个生成元，则 $o(a^r)=m\Leftrightarrow\frac{m}{(r,m)}=m\Leftrightarrow(r,m)=1$。

2.1.2　正规子群与商群

定义 2.1.5(等价关系)　$R\subset S\times S$ 称为等价关系，如果：

1）$(s,s)\in R$（自反性）；

2）$(s,t)\in R \Rightarrow (t,s)\in R$（对称性）；

3）$(s,t),(t,u)\in R \Rightarrow (s,u)\in R$（传递性）；

$(s,t)\in R$ 有时也写成 sRt。

定义 2.1.6 假设 n 是一个正整数。对任何整数 a、b，如果 $n\mid(a-b)$，则称 a 和 b 模 n（或 mod n）同余，记做 $a\equiv b \bmod n$。n 称为这个同余式的模。

显然模 n 的同余关系是整数集合 $\mathbf{Z}$ 上的一个等价关系，并将 $\mathbf{Z}$ 分成了 n 个互不相交的等价类$[0],[1],[2],\cdots,[n-1]$，每个等价类都称为模 n 的剩余类。

例 2.1.6 $\{[0],[1],[2],\cdots,[n-1]\}$在运算$[a]+[b]=[a+b]$下构成一个加法群，称为模 n 的剩余类群，记做 Z_n。

定义 2.1.7 群 G 的一个子集 H 称为一个子群，如果在 G 的运算下，H 构成一个群。

定理 2.1.5 设 G 是群，对任何的 $a\in G$，定义$\langle a\rangle=\{a^i \mid i\in\mathbf{Z}\}$，则$\langle a\rangle$是 G 的子群。且如果$\langle a\rangle$是有限群，则$\langle a\rangle$的阶恰好等于 a 的阶。

定理 2.1.6 如果 H 是群 G 的一个子群，则 G 上的关系 R_H：$(a,b)\in R_H \Leftrightarrow a=bh$（对某个 $h\in H$）是一个等价关系。

证明：要证 R_H 是个等价关系，只要验证它满足定义 2.1.5 中的 3 条性质即可。

1）$(a,a)\in R_H$，因为单位元 $1\in H$。

2）$(a,b)\in R_H \Rightarrow a=bh, h\in H \Rightarrow b=ah^{-1} \Rightarrow (b,a)\in R_H$。

3）$(a,b)\in R_H,(b,c)\in R_H \Rightarrow a=bh_1, b=ch_2 \Rightarrow a=ch_2h_1=c(h_2h_1) \Rightarrow (a,c)\in R_H$。

上述关系称为模 H 的左同余，同样有模 H 的右同余关系，等价类 aH 或 Hb 称为 H 在 G 中的左陪集（left coset）或右陪集（right coset）。若 $aH=Ha$，则称 aH 为 H 在 G 中的陪集。

例 2.1.7 $G=Z_{12}$，$H=\{[0],[3],[6],[9]\}$，则 H 的陪集有：

$$[0]+H=\{[0],[3],[6],[9]\}$$
$$[1]+H=\{[1],[4],[7],[10]\}$$
$$[2]+H=\{[2],[5],[8],[11]\}$$

定理 2.1.7 如果 H 是 G 的一个有限子群，则 H 每一个（左或右）陪集都和 H 有同样多的元素。

定义 2.1.8 如果群 G 的子群 H 只构造出有限多个模 H 的陪集，则这个陪集的个数称为 H 在 G 中的指数。

定理 2.1.8（Lagrange） 一个有限群 G 的阶正好等于任何一个子群 H 的阶乘以 H 在 G 中的指数。特别地，H 的阶整除群 G 的阶，任一元素的阶整除 G 的阶。

证明：由于 G 是有限的，所以 H 的陪集的个数 j 也是有限的。由于每一个陪集都和 H 含有相同个数的元素，且两个不同的陪集互不相交，所以$|G|=|H|j$。

例 2.1.8（欧拉（Euler）定理） 设 n 是一正整数，考察由模 n 的等价类构成的集合 $G=\{[a]\mid(a,n)=1\}$，则 G 在模 n 的乘法运算下构成一有限群，阶为$|G|=\phi(n)$。对任给 $a\in\mathbf{Z}$，若$(a,n)=1$，则$[a]\in G$，所以$[a]$的阶是$|G|$的因子，因此$[a]^{\phi(n)}=1$，也

就是 $a^{\phi(n)}\equiv 1(\mathrm{mod}\ n)$。

在上述例子中，如果取 n 为某个素数 p，则得到费尔马(Fermat)定理：设 p 是一素数，则对任意 $a\neq 0, a^{p-1}\equiv 1(\mathrm{mod}\ p)$。

定义 2.1.9 群 G 的一个子群 H 称为正规子群，如果对任何 $a\in G, h\in H$，有 $aha^{-1}\in H$。如果 H 是 G 的正规子群，则记成 $H\lhd G$。

显然，交换群的任一子群都是正规子群。

定理 2.1.9 设 H 是群 G 的子群，则下列条件彼此等价：

1) $H\lhd G$;

2) 对于每个 $g\in G, gHg^{-1}=H$;

3) H 的每个左陪集都是右陪集。事实上，对于每个 $g\in G, gH=Hg$。

证明：1) ⇒2)：假设 H 是 G 的正规子群，g 是任一给定的 G 中的元素。根据正规子群的定义，易知 $gHg^{-1}\subset H$。另外，$\forall h\in H$，因为 $g^{-1}\in G$，所以根据正规子群的定义，也有 $(g^{-1})h(g^{-1})^{-1}\in H$，所以存在 $h'\in H$ 使 $(g^{-1})h(g^{-1})^{-1}=h'$，因此 $h=gh'g^{-1}$，$H\subset gHg^{-1}$。从而 $H\subset gHg^{-1}$.

2) ⇒3)：当 $gHg^{-1}=H$ 时，$gH=Hg$ 是显然的。

3) ⇒1)：假设 H 的每个左陪集都是右陪集，则对任何的 g，存在 g' 使得 $gH=Hg'$。由于 $1\in H$，因此有 $h'\in H$ 满足 $g=g\cdot 1=h'g'$，从而 $Hg=Hg'=gH$，所以对任何 $h\in H$，存在 $h''\in H$ 使得 $gh=h''g$，所以 $ghg^{-1}=h''\in H$。从而 H 是一个正规子群。

定理 2.1.10 如果群 G 的子群 H 是正规的，则模 H 的陪集的集合在运算 $(aH)\cdot(bH)=(ab)H$ 下构成一个群。

证明：关键是证明上述定义的运算是良定义的。即如果 $a_1\in aH, b_1\in bH$，则 $a_1b_1H=(ab)H$。由 H 正规可知 $a_1=ha, b_1=bh'$，所以 $a_1b_1=h(ab)h'=(ab)h''h'$（因为 H 是正规的），其中 $h,h',h''\in H$。所以，$a_1b_1H=abH$。

定义 2.1.10 设 H 是 G 的正规子群，定理 2.1.10 中由 H 的陪集定义的群称为 G 关于 H 的商群，记做 G/H。

定理 2.1.11 如果 G 是有限群，则 $|G/H|=|G|/|H|$。

证明：由定理 2.1.8 立得。

定义 2.1.11 设 S 是群 G 的非空子集，S 在 G 中的正规化子定义为 $N(S)=\{a\in G\,|\,aSa^{-1}=S\}$。

定理 2.1.12 对 G 的任一非空子集 S，$N(S)$ 是 G 的子群且 $N(S)$ 的左陪集和 S 的不同的共轭 aSa^{-1} 之间存在一一对应关系。

证明：显然，G 的单位元 $1\in N(S)$。设 $a,b\in N(S)$，容易推出 a^{-1} 和 ab 也属于 $N(S)$，所以 $N(S)$ 是子群。下面证定理的第二部分：考虑 $aSa^{-1}=bSb^{-1}\Leftrightarrow S=a^{-1}bSb^{-1}a=(a^{-1}b)S(a^{-1}b)^{-1}\Leftrightarrow a^{-1}b\in N(S)\Leftrightarrow b\in aN(S)$。这就是说，$S$ 的两个共轭相同当且仅当这两个共轭是由 $N(S)$ 的同一个陪集中的元素定义的。这样 $aSa^{-1}\leftrightarrow aN(S)$ 就是一一对应的。

定义 2.1.12 对任何的群 G，G 的中心(center)定义为集合 $C=\{c\in C\,|\,ac=ca$，

$\forall a\in G\}$。显然 C 是 G 的正规子群，且 G 是交换的当且仅当 $C=G$。

定理 2.1.13 设 G 是一个有限群，其中心为 C，则 $|G|=|C|+\sum_{i=1}^{k}n_i$。其中 $n_i\geqslant 2$ 且 $n_i\mid |G|$。事实上，$n_1,n_2,\cdots,n_k$ 是 G 中含有两个和两个以上元素的不同的共轭类中元素的个数。

证明：注意到 G 中元素之间的共轭关系是一个等价关系。所以可以把 G 分成一些不相交的共轭类的并，因此 $|G|$ 等于这些不同的共轭类中元素个数之和。只含有一个元素的共轭类共有 $|C|$ 个（C 中每一元素对应一个这样的类）。而 $n_1,n_2,\cdots,n_k$ 则是剩下的共轭类中元素的个数。要证明 $n_i\mid |G|$，只要注意到 n_i 是与某一个 $a\in G$ 共轭的元素的个数。由定理 2.1.12 知 n_i 应等于 $N(\{a\})$ 的陪集的个数。因而整除 $|G|$。

2.1.3 群的同态与同构

定义 2.1.13 设 $f: G\to H$ 是群 G 到群 H 的一个映射。如果 $\forall a,b\in G$，$f(a\cdot b)=f(a)\cdot f(b)$ 则称 f 为 G 到 H 的同态。进一步，如果 $f(G)=H$，则称 f 为满同态；如果 f 是单映射，则称 f 为单同态映射；如果 f 是一一对应，则称 f 为同构映射，这时称 G 和 H 是同构的，记做 $G\cong H$。G 到 G 自身的同构映射称为自同构。

同态映射总是把单位元映射到单位元，把逆元素映射成像的逆元素。

例 2.1.9（内自同构） 设 G 是群。$\forall a\in G$，定义 $f_a: f_a(b)=aba^{-1}, b\in G$。则 f_a 称为群 G 的由 a 定义的内自同构，而把一个元素在内自同构下的像称为该元素的共轭元。显然，群中元素的共轭关系是一个等价关系。

定理 2.1.14 一个群 G 的所有自同构映射在映射的复合运算下构成一个群，称为 G 的自同构群。

定义 2.1.14 设 $f: G\to H$ 是群同态映射。f 的核（kernel）定义为 $\ker f=\{a\in G\mid f(a)=1_H\}$。其中，$1_H$ 是 H 中的单位元。

例 2.1.10 定义映射 $f: Z\to Z_n: f(a)=[a]$，则 f 是同态映射，且 $\ker f=\langle n\rangle$。

定理 2.1.15（同态基本定理） 设 $f: G\to H$ 是群 G 到群 H 上的满同态，那么 $\ker f$ 是 G 的一个正规子群。而且 H 同构于商群 $G/\ker f$，即 $G/\ker f\cong H$。反之，如果 N 是 G 的正规子群，则映射 $\varphi: G\to G/N: \varphi(a)=aN$ 是 G 到 G/N 的满同态且 $\ker\varphi=N$。

证明：$\forall a\in G, h\in\ker f$，则 $f(aha^{-1})=f(a)\cdot f(h)\cdot f(a^{-1})=f(a)\cdot 1_H\cdot f(a^{-1})=f(a)f(a^{-1})=f(a\cdot a^{-1})=1_H$，所以 $aha^{-1}\in\ker f$。这样就证明了 $\ker f$ 是 G 的正规子群。

令 $N=\ker f$，定义映射 $\psi: G/\ker f\to H: aN\to f(a)$。因为若 $aN=bN$，则存在 $n'\in N$ 使 $a=bn'$，所以 $f(a)=f(bn')=f(b)f(n')=f(b)$。因此 ψ 是良定义的。下面就来证明 ψ 是从 $G/\ker f$ 到 H 的同构映射。ψ 是满射这是显然的，因此只需证明 ψ 是单的同态映射即可。

一方面，因为 $\psi(aN)=1_H\Rightarrow f(a)=1_H\Rightarrow a\in N$，所以 $aN=N$，即 aN 为单位元，因此 ψ 是单映射。另一方面，由于对任何的 $a,b\in G$，$\psi(aN\cdot bN)=\psi((a\cdot b)N)=$

$f(ab)$，$\psi(aN)\cdot\psi(bN)=f(a)\cdot f(b)=f(ab)$，因此 $\psi(aN\cdot bN)=\psi(aN)\cdot\psi(bN)$，所以 ψ 是同态映射。综上，证明了 ψ 是同构映射，因此 $G/\ker f\cong H$。

定理的另一部分结论可以直接验证。

2.2　环与理想

2.2.1　基本概念与基本原理

上节介绍了群的概念，它是一种只包含一个代数运算的代数结构，而在实际应用中，经常会碰到包含有多个运算的代数系统，如整数就包含有加法和乘法两种运算。这一节将要介绍包含有两种代数运算的代数结构。请看下面的定义。

定义 2.2.1　一个集合 R 称为一个环，如果 R 有一个加法（$+$）和一个乘法运算（$\cdot$）满足：

1）$(R,+)$ 是一个交换群；

2）乘法运算满足结合律，即 $(a\cdot b)\cdot c=a\cdot(b\cdot c)$；

3）加法和乘法满足分配律，即对任何 $a,b,c\in R$，$a(b+c)=ab+ac$，$(b+c)a=ba+ca$。

例 2.2.1　所有的整数在通常数的加法和乘法运算下形成一个整环，称为整数环，用 Z 表示。这是因为整数在加法运算下正好构成一交换群，而乘法运算和加法运算又满足分配律。

定义 2.2.2　1）一个环称为有单位元的，如果它有乘法单位元 1。

2）一个环称为交换环，如果其中的乘法运算是交换的。

3）一个环称为整环，如果它是一个交换的有单位元的环，$1\neq 0$ 且对于任意的 $a\neq 0,b\neq 0\Rightarrow ab\neq 0$。

4）一个环称为除环，如果所有非零元在乘法运算下构成一个群。

5）一个交换的除环称为域。

在一个环中，如果 $a\neq 0,b\neq 0$ 但 $ab=0$，则称 a,b 是零因子。按照上述定义，整环显然是一个没有零因子的有单位元的交换环。

例 2.2.2　1）设 $(R,+)$ 为一 Abel 群，对 $a,b\in R$，定义 $a\cdot b=0$，则 R 在原来的加法运算和新定义的乘法运算下构成一环。

2）所有偶数在通常数的加法运算和乘法运算下形成一个没有单位元的交换环。

3）所有从实数到实数的映射按照运算

$$(f+g)(x)=f(x)+g(x),\quad (fg)(x)=f(x)g(x),\quad \forall x\in R$$

形成一个有单位元的交换环。

4）一个数域 K 上的所有 n 阶方阵按照矩阵的加法和乘法构成有单位元的非交换环。

5）所有的有理数在通常数的加法运算和乘法运算下构成一域，称为有理数域，记做 Q。

定义 2.2.3 环 R 的一个非空子集 S 称为 R 的一个子环，如果 S 关于 $+$ 和 $\cdot$ 是封闭的并且在这两种运算下形成一个环。

定义 2.2.4 环 R 的一个非空子集 J 称为一个理想，如果 J 是 R 的一个子环并且对所有 $a\in J, r\in R$，有 $ar\in J$ 和 $ra\in J$。

显然任何一个环一定包含两个平凡的理想，一个是由整个环组成的理想，另一个是由单个 0 元素组成的零理想。非平凡的理想称为环的真理想。

例 2.2.3 设 Q 为有理数域，则整数集 Z 是 Q 的子环，但不是理想。因为 $1\in Z$，$\frac{1}{2}\in Q$，但是 $\frac{1}{2}\cdot 1=\frac{1}{2}\notin Z$。

例 2.2.4 设 R 为交换环，$a\in R$，则包含 a 的最小理想 $(a)=\{ra+na: r\in R, n\in Z\}$，这个理想称为由 a 生成的理想。特别地，如果 R 包含一个单位元，则 $(a)=\{ra: r\in R\}$。

定义 2.2.5 R 是一个交换环，R 的一个理想 J 称为主理想，如果存在 $a\in R$ 使得 $J=(a)$。

定义 2.2.6 设 R 是一整环。如果 R 中的每一个理想都是主理想，则称 R 是主理想整环。

例 2.2.5 所有的整数在通常数的加法运算和乘法运算下形成的整数环 Z 是主理想整环。事实上，任给 $J\subset Z$ 是 Z 的一个理想，如果 $J=\{0\}$，则 J 已经是一个主理想。现在假设 $J\neq\{0\}$。因为如果 $b\in J$，那么 $-b\in J$，所以 J 中一定有非负整数。令 d 是 J 中最小的非负整数，我们来证 $J=(d)$。

任取 $a\in J$，则存在 t 和 $0\leqslant r<d$ 使 $a=td+r$，所以 $r=a-td\in J$。但由于 d 是 J 中最小的正整数，因此只能有 $r=0$。所以 $a=td\in(d)$，从而 $J=(d)$。

因为环的理想可以作为环的加法群的正规子群，所以环 R 的一个理想 J 将环 R 分成一些互不相交的陪集的并，每个陪集叫模 J 的剩余类。把元素 a 所在的剩余类记为 $[a]=a+J$（因为里面的元素都是具有形式 $a+c, c\in J$）。

可以直接验证，环 R 模 J 的剩余类按运算：

$$(a+J)+(b+J)=(a+b)+J$$

$$(a+J)(b+J)=ab+J$$

可以形成一个环。

定义 2.2.7 按上述运算所作成的由剩余类构成的环称为 R 模 J 的剩余类环（或商环），记做 R/J。

定义 2.2.8 设 R, S 为环，$a, b\in R$。一个映射 $\varphi: R\to S$ 称为环同态，如果：

1）$\varphi(a+b)=\varphi(a)+\varphi(b)$；

2）$\varphi(ab)=\varphi(a)\varphi(b)$。

而集合 $\ker\varphi=\{a\in R: \varphi(a)=0\in S\}$ 称为同态映射 φ 的核。一个同态映射，如果既是单的又是满的，则称为同构映射。如果两个环之间存在同构映射，则说这两个环是同构的，用 $\simeq$ 表示。

类似于群的同态基本定理，也有以下定理。

定理 2.2.4（同态基本定理） 设 R 和 S 是环。如果 $\varphi: R\to S$ 是满同态，则 $\ker\varphi$

为 R 的理想，且 $S\cong R/\ker\varphi$。反过来，如果 J 为 R 的理想，定义映射 $\varphi(a)=a+J$，$a\in R$，则映射 $\varphi: R\to R/J$ 是满同态且 $\ker\varphi=J$。

证明：作为练习，此处省略。

例 2.2.6　设 n 是一个正整数，(n)表示由 n 生成的整数环 Z 中的理想。那么可以得到模(n)的剩余类环 $Z/(n)$，有时也把这个环称为模 n 的剩余类环，并用 Z_n 表示。

例 2.2.7　容易验证，当 p 是素数时，$Z_p=Z/(p)$是一域，称为阶为 p 的 Galois 域，并用 F_p 表示。有时也把 Z_p 中的元素简单写为 $0,1,2,\cdots,p-1$。

定义 2.2.9　设 R 是一个环，对 R 中的一个真理想 P，如果 $ab\in P\Rightarrow a\in P$ 或 $b\in P$，则称 P 为 R 的素理想。

定义 2.2.10　设 M 是环 R 的一个真理想。如果对 R 的任意一个理想 J，$M\subset J$，一定有 $J=R$ 或 $J=M$，则称 M 是 R 的极大理想。

定理 2.2.2　设 R 是一个有单位元的交换环，则

1）理想 M 是极大理想$\Leftrightarrow R/M$ 是域。

2）理想 P 是素理想$\Leftrightarrow R/P$ 是整环。

3）每个极大理想都是素理想。

证明：

1）$\Rightarrow$：设 M 是极大理想。$\forall a\notin M$，则 $J=\{ar+m\mid r\in R,m\in M\}$是一个理想且 $J\supsetneq M$，从而 $J=R$。所以 $\exists r\in R,m\in M$ 使得 $ar+m=1$，因此在 R/M 中，$[a]\cdot[r]=[1]$。所以，R/M 是域。

$\Leftarrow$：假设 R/M 是域，$J\supseteq M$ 且 $J\neq M$，则存在 $a\in J,a\notin M$，所以$[a]\neq 0$。设$[a]$的逆元为$[r]$，即$(a+M)(r+M)=1+M$，那么 $ar+m=1$ 对某个 $m\in M$ 成立。又因为 $m\in M\subseteq J$，$a\in J$，所以 $1=ar+m\in J$，从而 $J=R$。

2）$\Rightarrow$：假设 P 是素理想，则 R/P 是一个交换环。单位元为$[1]=1+P\neq 0+P$。假设$(a+P)(b+P)=0+P$，则 $ab+P=0+P$，所以 $ab\in P$，从而 $a\in P$ 或 $b\in P$ 即$[a]=0$ 或$[b]=0$。所以 R/P 没有零因子。

$\Leftarrow$：设 $ab\in P$，则$[a][b]=[ab]=0$。所以$[a]=0$ 或$[b]=0$，即 $a\in P$ 或 $b\in P$。

3）由 1）和 2）立得。

定理证毕。

2.2.2　多项式环

设 R 是环，$R[x]$是由系数在 R 中，未定元为 x 的多项式的全体组成的集合，那么任给 $R[x]$中的一个非零元素 $f(x)\in R[x]$，一定可以写成

$$f(x)=a_nx^n+\cdots+a_1x+a_0,\quad a_n\neq 0$$

的形式，其中 $a_0,a_1,\cdots,a_n\in R$ 且 $a_n\neq 0$。n 称为多项式 $f(x)$的次数，记为 $\deg(f(x))$。首先在 $R[x]$中引入加法运算。设 $f(x)=a_nx^n+\cdots+a_1x+a_0$，$g(x)=b_nx^n+\cdots+b_1x+b_0$ 是 $R[x]$中的两个多项式，定义 $f(x)$和 $g(x)$的加法为

$$f(x)+g(x)=(a_n+b_n)x^n+\cdots+(a_1+b_1)x+(a_0+b_0)$$

其次，再定义 $R[x]$ 中多项式的乘法运算。设 $f(x)=a_m x^m+\cdots+a_1 x+a_0, a_m\neq 0$ 和 $g(x)=b_n x^n+\cdots+b_1 x+b_0, b_n\neq 0$ 是 $R[x]$ 中的两个次数分别为 m 和 n 的多项式，定义 $f(x)$ 和 $g(x)$ 的乘积为

$$f(x)\cdot g(x)=c_{m+n}x^{m+n}+c_{m+n-1}x^{m+n-1}+\cdots+c_1 x+c_0$$

其中 $c_k=\sum_{0\leqslant i\leqslant n,0\leqslant j\leqslant m,i+j=k}a_i b_j, k=0,1,\cdots,m+n$，即

$$c_{m+n}=a_m b_n, c_{m+n-1}=a_m b_{n-1}+a_{m-1}b_n,\cdots,c_1=a_1 b_0+a_0 c_1, c_0=a_0 b_0$$

显然，$R[x]$ 在上面定义的运算下构成一个环，这个环称为环 R 上的单变元多项式环。同样，可以递归地定义 R 上的多变元多项环 $R[x_1,x_2,\cdots,x_n]$：

$$R[x_1,x_2,\cdots,x_n]=R[x_1][x_2]\cdots[x_n]$$

其中，$x_1,x_2,\cdots,x_n$ 是未定元。当 $n=1$ 时，多变元多项式环就是通常的单变元多项式环。

对于环 R 上的多项式环，有以下定理。

定理 2.2.3 设 $R[x_1,\cdots,x_n]$ 是环 R 上的多项式环，那么

1) $R[x_1,\cdots,x_n]$ 是交换的 $\Leftrightarrow R$ 是交换的。

2) $R[x_1,\cdots,x_n]$ 有单位元 $\Leftrightarrow R$ 有单位元。

3) $R[x_1,\cdots,x_n]$ 是一个整环 $\Leftrightarrow R$ 是整环。

在这一节，主要讨论域上的单变元多项式环，而把系数在域 F 中的多项式称为域 F 上的多项式。从上面的定理容易看出，对于一个域 F，域上的多项式环 $F[x]$ 一定是一个整环。首先看下面的定义。

定义 2.2.11 设 $f(x)$ 和 $g(x)$ 是域 F 上的两个任意多项式，$g(x)\neq 0$。如果存在一个多项式 $q(x)\in F[x]$ 使得

$$f(x)=q(x)g(x)$$

成立，那么就称 $g(x)$ 整除 $f(x)$ 或者 $f(x)$ 可以被 $g(x)$ 整除，记做 $g(x)\mid f(x)$。这时把 $g(x)$ 叫做 $f(x)$ 的因式，而把 $f(x)$ 叫做 $g(x)$ 的倍式。显然，任何一个多项式，一定有两种因式，一种是非零的常数多项式，另一个就是多项式本身，我们把这两种因式称为多项式的平凡因式，否则称为非平凡因式。如果一个多项式不能写成两个非平凡因式的乘积，那么就称这个多项式是不可约多项式，或既约多项式。

不可约多项式的概念是多项式环中的一个重要概念，从本节后面的定理可以看到 $F[x]$ 中的每一个多项式都可以分解成不可约多项式的乘积。

定理 2.2.4（多项式除法） $g(x)\in F[x], g(x)\neq 0$，则对任何 $f(x)\in F[x]$，一定存在多项式 $q(x), r(x)\in F[x]$，满足

$$f(x)=g(x)\cdot q(x)+r(x),\quad \deg(r(x))<\deg(g(x))$$

证明：如果 $\deg(f(x))<\deg(g(x))$，则定理显然成立，这时只要取 $q(x)=0$，$r(x)=f(x)$ 即可。现在假设 $\deg(f(x))\geqslant\deg(g(x))$。用数学归纳法来证明定理仍然成立。

假设对 $\deg(f(x))<n$ 的多项式 $f(x)$ 定理成立，现在来考察 $\deg(f(x))=$

$n \geqslant \deg(g(x))$的情况。设 $f(x)=a_n x^n+\cdots+a_1 x+a_0$, $g(x)=b_m x^m+\cdots+b_1 x+b_0$, $n \geqslant m$。因为

$$f(x)-b_m{}^{-1}a_n x^{n-m}g(x)=(a_{n-1}-b_m{}^{-1}a_n b_{m-1})x^{n-1}+\cdots+\text{低次项}$$

所以 $f(x)-b_m{}^{-1}a_n x^{n-m}g(x)$是次数不大于 $n-1$ 的多项式。根据归纳假设,知存在多项式 $q_1(x)$和 $r(x)$使得

$$f(x)-b_m{}^{-1}a_n x^{n-m}g(x)=g(x)q_1(x)+r(x),\quad \deg(r(x))<\deg(g(x))$$

所以

$$f(x)=(q_1(x)+b_m{}^{-1}a_n x^{n-m})g(x)+r(x)=q(x)g(x)+r(x)$$

其中,$q(x)=q_1(x)+b_m{}^{-1}a_n x^{n-m}$。定理得证。

例 2.2.8 $f(x)=2x^5+x^4+4x+3 \in F_5[x]$, $g(x)=3x^2+1 \in F_5[x]$。因此有

$$f(x)=(4x^3+2x^2+2x+1)g(x)+(2x+2)$$

即 $q(x)=4x^3+2x^2+2x+1$, $r(x)=2x+2$。

定理 2.2.5 设 F 是域,那么 F 上的多项式环 $F[x]$是主理想整环,即对任何 $J \neq (0)$是 $F[x]$中的一个理想,必定存在多项式 $g \in F[x]$使得 $J=(g)$。

证明:因为 $J \neq (0)$,因此 J 中一定存在非零多项式。假设 $g(x)$是 J 中一个次数最低的多项式,下面证明 g 是 J 的一个生成元。

设 $f(x)$是 J 中的任意一个元素,则根据定理 2.2.4 一定存在 $q(x), r(x) \in F[x]$,使得

$$f(x)=q(x)g(x)+r(x),\quad \deg(r(x))<\deg(g(x))$$

所以 $r(x)=f(x)-q(x)g(x)$。注意到 J 是理想,$g(x) \in J$,因此 $q(x)g(x) \in J$,从而 $r(x)=f(x)-q(x)g(x) \in J$,而且 $\deg(r(x))<\deg(g(x))$。但是根据假设,$g(x)$是 J 中次数最低的多项式,因此只能有 $r(x)=0$,也就是说 $f(x)=q(x)g(x)$,所以 $J=(g(x))$。定理得证。

定理 2.2.6 假设 $f_1, f_2, \cdots, f_n$ 是 $F[x]$中不全为零的一组多项式,则存在唯一的首一多项式 d 满足:

1) d 整除每一个 f_i。

2) 整除每个 f_i 的 c 一定整除 d。更进一步,$\exists b_i \in F[x]$使得 $d=\sum b_i f_i$。

证明:令 $J=\left\{\sum c_i f_i \mid c_i \in F[x]\right\}$,则 J 是 $F[x]$中的理想,且 $f_i \in J$。由于 $F[x]$是主理想整环,所以存在首一多项式 $d \in F[x]$,使 $J=(d)$。所以 $d \mid f_i$ 且 $d=\sum c_i f_i$(因为 $d \in J$)。所以 2)成立。

假设有另一个 d_1 满足 1)和 2),则 d_1 和 d 是相伴的,因此 $d_1=d$。

上述定理中的首一多项式 d 称为 $f_1, \cdots, f_n$ 的最大公因式,表示为 $d=\gcd(f_1, \cdots, f_n)$或 $d=(f_1, \cdots, f_n)$。如果 $d=1$,则称 $f_1, \cdots, f_n$ 是互素的。

推论 2.2.1 假设 $f_1, f_2, \cdots, f_n$ 是 $F[x]$中不全为零的一组多项式。如果多项式 $d(x) \in F[x]$是 $f_1, \cdots, f_n$ 的最大公因式,那么一定存在一组多项式 $c_1(x), \cdots, c_n(x) \in F[x]$,使得

$$d(x)=c_1(x)f_1(x)+\cdots+c_n(x)f_n(x)$$

推论 2.2.2 设 $f(x),g(x),h(x)\in F[x]$，$h(x)\mid f(x)g(x)$。那么，如果 $(h(x),f(x))=1$，则 $h(x)\mid g(x)$。

证明： 根据推论 2.2.1，存在多项式 $s(x),t(x)$ 使

$$s(x)f(x)+h(x)t(x)=1$$

两边同乘以 $g(x)$ 得

$$\begin{aligned}g(x)&=s(x)f(x)g(x)+h(x)t(x)g(x)\\&=s(x)\cdot(f(x)g(x))+h(x)\cdot(t(x)g(x))\end{aligned}$$

因为 $h(x)$ 整除上式右边的每一项，因此 $h(x)$ 也整除上式的左边，即 $h(x)\mid g(x)$。

下面以定理的形式给出求两个多项式的最大公因式的辗转相除法。

定理 2.2.7（辗转相除法） 设 $f(x),g(x)$ 是域 F 上的两个多项式，$g(x)\neq 0$。记 $r_0(x)=f(x)$，$r_1(x)=g(x)$，并反复使用定理 2.2.4 给出的多项式除法，则有

$$\begin{aligned}r_0(x)&=q_1(x)r_1(x)+r_2(x) && 0\leqslant\deg(r_2(x))<\deg(r_1(x))\\r_1(x)&=q_2(x)r_2(x)+r_3(x) && 0\leqslant\deg(r_3(x))<\deg(r_2(x))\\&\vdots\\r_{k-1}(x)&=q_k(x)r_k(x)+r_{k+1}(x)\\& && 0\leqslant\deg(r_{k+1}(x))<\deg(r_k(x))\\&\vdots\end{aligned}$$

上述过程经过有限步后，一定存在 k 使得 $r_{k+1}=0$。这时得到的 $r_k(x)$ 就是多项式 $f(x),g(x)$ 的最大公因式，即 $r_k(x)=(f(x),g(x))$。

实际上，利用上述定理给出的算法，也可以求出推论 2.2.1 中最大公因式的表达式，这只要从算法中把 $r_2(x),r_3(x),\cdots,r_{k-1}$ 逐次消去即可。

例 2.2.9 设 $f(x)=x^{13}+x^{11}+x^9+x^8+x^6+x^5+x^4+x^3+1$，$g(x)=x^8+x^4+x^3+x+1\in F_2[x]$，求多项式 $s(x),t(x)\in F_2[x]$ 使

$$(f(x),g(x))=s(x)f(x)+t(x)g(x)$$

令 $r_0(x)=f(x)$，$r_1(x)=g(x)$，运用辗转相除法，则有

$$\begin{aligned}&r_0(x)=q_1(x)r_1(x)+r_2(x), && q_1(x)=x^5+x^3, && r_2(x)=x^7+x^6+1,\\&r_1(x)=q_2(x)r_2(x)+r_3(x), && q_2(x)=x+1, && r_3(x)=x^6+x^4+x^3,\\&r_2(x)=q_3(x)r_3(x)+r_4(x), && q_3(x)=x+1, && r_4(x)=x^5+x^3+1,\\&r_3(x)=q_4(x)r_4(x)+r_5(x), && q_4(x)=x, && r_5(x)=x^3+x,\\&r_4(x)=q_5(x)r_5(x)+r_6(x), && q_5(x)=x^2, && r_6(x)=1,\\&r_5(x)=q_6(x)r_6(x), && q_6(x)=x^3+x,\end{aligned}$$

因此，$(f(x),g(x))=r_6(x)=1$，且

$$\begin{aligned}r_6(x)&=r_4(x)+q_5(x)r_5(x)\\&=r_4(x)+q_5(x)(r_3(x)+q_4(x)r_4(x))\\&=(x^2)r_3(x)+(1+x^3)r_4(x)\\&=(x^2)r_3(x)+(1+x^3)(r_2(x)+q_3(x)r_3(x))\\&=(1+x^3)r_2(x)+(x^4+x^3+x^2+x+1)r_3(x)\end{aligned}$$

$$= (1 + x^3)r_2(x) + (x^4 + x^3 + x^2 + x + 1)(r_1(x) + q_2(x)r_2(x))$$
$$= (x^4 + x^3 + x^2 + x + 1)r_1(x) + (x^5 + x^3)r_2(x)$$
$$= (x^4 + x^3 + x^2 + x + 1)r_1(x) + (x^5 + x^3)(r_0(x) + q_1(x)r_1(x))$$
$$= (x^5 + x^3)r_0(x) + (x^{10} + x^6 + x^4 + x^3 + x^2 + x + 1)r_1(x)$$
$$= (x^5 + x^3)f(x) + (x^{10} + x^6 + x^4 + x^3 + x^2 + x + 1)g(x)$$

因此，$s(x)=x^5+x^3$，$t(x)=x^{10}+x^6+x^4+x^3+x^2+x+1$。

相对于多项式的最大公因式，也可以定义多项式的最小公倍式，请看下面的定义。

定义 2.2.12　设 $f(x)$，$g(x)$是域 F 上的两个多项式。$m(x)\in F[x]$是域 F 上的另一多项式。如果

(1) $f(x) \mid m(x)$，$g(x) \mid m(x)$；

(2) 对 $h(x)\in F[x]$，如果也有 $f(x)|h(x)$，$g(x)|h(x)$，则 $m(x)|h(x)$。

那么 $m(x)$就叫做 $f(x)$，$g(x)$的最小公倍式，记做$[f(x),g(x)]$或 $\mathrm{lcm}(f(x),g(x))$。

定理 2.2.8　设 $p(x)$是域 F 上的一个不可约多项式，即 $p(x)\in F[x]$。那么由 $p(x)$生成的理想$(p(x))$是 $F[x]$中的极大理想，因此也是 $F[x]$中的素理想。

证明：假设 J 是 $F[x]$的一个理想，且 $J\supseteq(p(x))$。需要证明如果 $J\neq(p(x))$，那么 $J=F[x]$。设有 $r(x)\in J$，但 $r(x)\notin(p(x))$。因为 $p(x)$是一个不可约多项式，所以$(p(x),r(x))$或者$=1$ 或者$=p(x)$。但由于 $r(x)\notin(p(x))$，因此 $p(x)\nmid r(x)$，从而$(p(x),r(x))\neq p(x)$，所以有$(p(x),r(x))=1$。根据推论 2.2.7，存在多项式 $s(x)$，$t(x)\in F[x]$，使得 $s(x)p(x)+t(x)r(x)=1$。但 $p(x)$，$r(x)\in J$，因此 $1=s(x)p(x)+t(x)r(x)\in J$，所以 $J=F[x]$。定理证毕。

推论 2.2.3　设 $p(x)$是域 F 上多项式环 $F[x]$中的一个多项式，则商环 $F[x]/(p(x))$是域当且仅当 $p(x)$是一个不可约多项式。

定理 2.2.9　设 F 是域，$p(x)$，$f_1(x)$，$f_2(x)$，…，$f_m(x)\in F[x]$且 $p(x)$是不可约的。如果

$$p(x) \mid f_1(x)f_2(x)\cdots f_m(x)$$

那么 $p(x)$一定整除至少其中的一个因式。

证明：考虑商环 $Q=F[x]/(p(x))$。因为 $p(x)$是不可约多项式，因此$(p(x))$是 $F[x]$中的极大理想，所以 Q 是域。用$[f_j(x)]$表示 Q 中 $f_j(x)$ $(j=1,2,\cdots,m)$所在的等价类，由题设知在 Q 中$[f_1(x)]\cdot[f_2(x)]\cdot\cdots\cdot[f_m(x)]=[f_1(x)f_2(x)\cdots f_m(x)]=[0]$。所以至少有一个$[f_j(x)]=[0]$，即 $f_j(x)\in(p(x))$。所以 $p(x)|f_j(x)$。

鉴于上述定理的结果，因此有时也把不可约多项式称为素多项式，它实际上是环 $F[x]$中的一个素元。

定理 2.2.10(唯一分解)　设 $f(x)$是域$\mathbb{F}$上多项式环 $F[x]$中的一个正次数多项式。那么 $f(x)$一定可以写成

$$f(x) = a(p_1(x))^{e_1}\cdots(p_k(x))^{e_k}$$

其中 $a\in\mathbb{F}$，$p_i(x)$是不同的不可约多项式，$e_i\geqslant1$，而且在不计次序的情况下，这种分

解是唯一的。

证明：由定理 2.2.9 立得，此处略。

2.3 域和扩域

设 F 是域，K 是 F 的子集。如果 K 在 F 的运算下也构成一个域，则称 K 为 F 的子域。而 F 则称为 K 的扩域（或扩张）。特别地，如果 $K \neq F$，则称 K 为 F 的真子域。

定义 2.3.1 一个域如果不包含任何真子域，则称为素域。如果一个域 F 的子域作为域是素域，则称该子域为 F 的素子域。

注意到任意多个子域的交仍然是子域，可以知道一个域的素子域实际上就是该域的所有子域的交。

例 2.3.1 显然，有理数域 Q 和阶为素数 p 的 Galois 域 F_p（参看例 2.2.7）都是素域。

定理 2.3.1 一个域 F 的素子域在特征为 p 时同构于阶为 p 的 Galois 域 F_p，而在特征为 0 时，同构于有理数域 Q。

证明：设 P 是 F 的素子域，则 P 一定包含 0 和 1。下面分特征为 0 和特征为 p 两种情况来证明。

$ch(F)=p$ 的情况：因为 $\{0,1\}\subset P$，所以 $\{m\cdot 1 \mid m\in\mathbb{Z}\}\subset P$。构造映射

$$\phi:\mathbb{Z}\to P: m\mapsto m\cdot 1$$

容易验证 ϕ 是一个环同态映射，且 $\ker\phi=(p)$。所以 $F_p=Z/(p)=Z/\ker\phi\cong\phi(Z)\subset P$。但由于 F_p 是域，P 又没有真子域，因此 $F_p\cong\phi(Z)=P$。

$ch(F)=0$ 的情况：因为 $\{0,1\}\subset P$，所以 $\{(m\cdot 1)(n\cdot 1)^{-1}\mid m,n\in Z\}\subset P$。构造映射：

$$\phi: Q\to P: m/n\mapsto(m\cdot 1)(n\cdot 1)^{-1}$$

容易验证 ϕ 是一个环的单同态映射。因此，$Q\cong\phi(Q)\subset P$。但由于 Q 是域，且 P 又没有真子域，因此 $Q\cong\phi(Q)=P$。定理证毕。

定义 2.3.2 K 是 F 的子域。M 是 F 的任何子集。$K(M)$ 定义为所有含有 M 和 K 的子域的交，称为添加 M 中的元素得到的 K 的扩域（或扩张）。显然 $K(M)$ 是含有 K 和 M 的最小的子域。当 $M=\{\theta_1,\theta_2,\cdots,\theta_n\}$ 时，记 $K(M)=K(\theta_1,\theta_2,\cdots,\theta_n)$。特别 $K(\theta)$ 称为单扩域（或单扩张），θ 称为 $K(\theta)$ 在 K 上的定义元。

定义 2.3.3 设 α 是域 F 的某一扩域中的元素。则称 α 是多项式 $f(x)\in F[x]$ 的一个根（零点）或者说 α 满足多项式 $f(x)$，如果 $f(\alpha)=0$。

定理 2.3.2（余数定理） 设 $f(x)\in F[x]$，$\alpha\in F$。那么 α 是 $f(x)$ 的一个根当且仅当 $(x-\alpha)\mid f(x)$。

证明：假设 $\alpha\in F$ 是 $f(x)$ 的一个根。根据多项式的除法，知道存在 $q(x)$，$r(x)\in F[x]$ 使得

$$f(x) = q(x)(x-\alpha)+r(x)$$

且 $\deg(r(x))<\deg(x-\alpha)=1$。将 $x=\alpha$ 代入上式得 $r(\alpha)=f(\alpha)=0$，但 $\deg(r(x))<1$，$r(x)$ 是一常数多项式，因此 $r(x)=0$。这样就证明了 $(x-\alpha)\mid f(x)$。

反过来，如果 $(x-\alpha)\mid f(x)$，则存在 $q(x)\in F[x]$，使 $f(x)=q(x)(x-\alpha)$，所以 $f(\alpha)=0$，因此 α 是 $f(x)$ 的一个根。定理证毕。

从上面的定理可以看出，α 是 $f(x)$ 的一个根和多项式 $x-\alpha$ 能整除 $f(x)$ 是等价的。如果存在整数 $m\geqslant 2$ 使 $(x-\alpha)^m\mid f(x)$，但 $(x-\alpha)^{m+1}\nmid f(x)$，则称 α 是 $f(x)$ 的 m 次重根，否则称 α 为 $f(x)$ 的单根。

定义 2.3.4　设 K 是 F 的一个子域，$\theta\in F$，如果 θ 满足 K 上的一个非零多项式，则称 θ 为 K 上的代数元。不是代数元的元素称为超越元。如果 K 的一个扩张中的每个元素都是 K 上的代数元，则该扩张称为代数扩张。

定义 2.3.5（极小多项式）　设 K 是 F 的一个子域，α 是 F 中域 K 上的一个代数元。那么 $K[x]$ 中满足 $g(\alpha)=0$ 的次数最小的多项式

$$g(x)=x^n+a_{n-1}x^{n-1}+\cdots+a_0$$

叫做 α 在 K 上的极小多项式，多项式 $g(x)$ 的次数称为代数元次数。

定理 2.3.3　如果 $\theta\in F$ 是 K 上的代数元，那么 θ 的极小多项式 g 具有以下性质：

1）g 是不可约的。

2）令 $J=\{f(x)\in K[x]\mid f(\theta)=0\}$，则 J 是 $K[x]$ 中的一个理想，且 $J=(g)$。从而，$K[x]$ 中多项式 $f(x)$ 满足 $f(\theta)=0\Leftrightarrow g(x)\mid f(x)$。

证明：

1）假设 $g(x)=h_1(x)h_2(x)$，其中 $1\leqslant\deg(h_1(x)),\deg(h_2(x))<\deg(g(x))$。那么 $h_1(\theta)h_2(\theta)=g(\theta)=0$，所以或者 $h_1(\theta)=0$，或者 $h_2(\theta)=0$。无论哪种情况都与 $g(x)$ 是 θ 的极小多项式相矛盾。

2）只需证明 J 是 $K[x]$ 中的理想即可。注意到如果 $f(x)$ 以 θ 为根，那么对任何的 $h(x)\in K[x]$，$f(x)h(x)$ 也以 θ 为根。不难验证 J 是 $K[x]$ 中的理想。定理证毕。

设 L 是域 K 的扩域，那么 L 可以看成 K 上的向量空间，这只要将 K 中的元素看作数，把 L 中的元素看成向量，K 中元素与 L 中元素在 L 中的乘法看作是数与向量的乘法即可。如果 L 作为 K 上的向量空间是有限维的，则称 L 是 K 的有限扩张，向量空间的维数称为扩张次数，记为 $[L:K]$。

定理 2.3.4　如果 L 是 K 的有限扩张，M 是 L 的有限扩张，则 $[M:K]=[M:L][L:K]$。

证明：设 $[M:L]=m$，$\alpha_1,\cdots,\alpha_m$ 是 M 在 L 上的一组基，$[L:K]=n$，$\beta_1,\cdots,\beta_n$ 是 L 在 K 上的一组基。那么对任意 $\alpha\in M$，α 可以写成 $\alpha=\gamma_1\alpha_1+\cdots+\gamma_m\alpha_m$，其中 $\gamma_i\in L$ $(1\leqslant i\leqslant m)$，而且每个 γ_i 又可以写成 $\gamma_i=r_{i1}\beta_1+\cdots+r_{in}\beta_n$ $(1\leqslant i\leqslant m)$，其中 $r_{ij}\in K$ $(1\leqslant i\leqslant m,1\leqslant j\leqslant n)$。所以

$$\alpha=\sum_{i=1}^{m}\gamma_i\alpha_i=\sum_{i=1}^{m}\Big(\sum_{j=1}^{n}r_{ij}\beta_j\Big)\alpha_i=\sum_{i=1}^{m}\sum_{j=1}^{n}r_{ij}\alpha_i\beta_j$$

如果能证明 mn 个元素 $\alpha_i\beta_j$ $(1\leqslant i\leqslant m,1\leqslant j\leqslant n)$ 在 K 上是线性无关的，那么定理就得

到了证明。假设

$$\sum_{i=1}^{m}\sum_{j=1}^{n} s_{ij}\alpha_i\beta_j = 0$$

其中 $s_{ij}\in K$。那么

$$\sum_{i=1}^{m}\Big(\sum_{j=1}^{n} s_{ij}\beta_j\Big)\alpha_i = 0$$

因此，从 $\alpha_1,\cdots,\alpha_m$ 在 L 上的线性无关性可得

$$\sum_{j=1}^{n} s_{ij}\beta_j = 0 \quad 1\leqslant i\leqslant n$$

再利用 $\beta_1,\cdots,\beta_n$ 在 K 上的线性无关性可推出所有的 $s_{ij}=0$。定理证毕。

定理 2.3.5 每个有限扩张都是代数扩张。

证明：假设 $[L:K]=m$，则 $\forall\theta\in L$，$m+1$ 个元素(向量)$1,\theta,\theta^2,\cdots,\theta^m$ 肯定是线性相关的。所以存在 $a_i\in K$，$\sum_{i=0}^{m} a_i\theta^i = 0$。因此 θ 满足多项式 $f(x) = \sum_{i=0}^{m} a_i x^i$，所以 θ 是代数元。

定理 2.3.6 设 θ 是域 K 上的代数元，极小多项式为 g，$\deg(g)=n$。那么：

1) $K(\theta)\cong K[x]/(g)$。

2) $[K(\theta):K]=n$，且 $\{1,\theta,\cdots,\theta^{n-1}\}$ 是 $K(\theta)$ 在 K 上的一组基。

3) 每一个 $\alpha\in K(\theta)$ 都是 K 上的代数元，其次数整除 n。

证明：

1) 定义 τ：$K[x]\to K(\theta)$：$f\to f(\theta)$，则 τ 是一同态映射，且 $\ker\tau=(g)$。所以 $\tau(K[x])\cong K[x]/(g)$，因此 $\tau(K[x])\subset K(\theta)$ 是子域。因为 $\theta\in\tau(K[x])$，所以也有 $K(\theta)\subset\tau(K[x])$，因此 $K(\theta)=\tau(K[x])$，从而 $K(\theta)\cong K[x]/(g)$。

2) 由于 $\tau(K[x])=K(\theta)$，所以 $\forall\alpha\in K(\theta)$，$\exists f\in K[x]$，使得 $f(\theta)=\alpha$。因为 $\deg(g)=n$，根据多项式除法可以要求 f 的次数小于 n。所以 α 可以写成 $1,\theta,\cdots,\theta^{n-1}$ 的组合。下面证明 $1,\theta,\cdots,\theta^{n-1}$ 是线性无关的。如果 $a_0+a_1\theta+\cdots+a_{n-1}\theta^{n-1}=0$，则知 θ 满足多项式 $f(x)=a_0+a_1x+\cdots+a_{n-1}x^{n-1}$。但由于 θ 是次数为 n 的代数元，因此只有 $f(x)=0$，从而推出 $a_0=a_1=\cdots=a_{n-1}=0$。所以，$[K(\theta):K]=n$ 且 $1,\theta,\cdots,\theta^{n-1}$ 是 $K(\theta)$ 在 K 上的一组基。

3) 因为 $K(\theta)$ 是 K 的有限扩张，根据定理 2.3.5，$K(\theta)$ 也是 K 的代数扩张，所以 $\alpha\in K(\theta)$ 是 K 上的代数元。进一步，$K(\theta)\supset K(\alpha)\supset K$，所以 $[K(\theta):K]=[K(\theta):K(\alpha)]\cdot[K(\alpha):K]$。所以 $[K(\alpha):K]$ 整除 n，也就是 α 的次数整除 n。

定理 2.3.7 设 $f(x)\in K[x]$ 是一个不可约多项式，则存在 K 的一个单代数扩张以 $f(x)$ 的一个根作为定义元。

证明：考察商环 $L=K[x]/(f(x))$。根据推论 2.2.3，L 是一域，其中的元素都是模 $(f(x))$ 的剩余类，即 $L=\{[h]=h+(f)\mid h\in K[x]\}$。构造从 K 到 L 的映射 ϕ：$a\mapsto[a]$。易知 ϕ 是一个同态单映射，所以 K 同构于 L 的某个子域，因此可以把 a 和 $[a]$ 等同起来，并把 L 看成 K 的一个扩域。而且，在上述对应下，对任一多项式 $h(x)=$

$a_0+a_1x+\cdots+a_mx^m\in K[x]$，有$[h]=[a_0+a_1x+\cdots+a_mx^m]=[a_0]+[a_1]x+\cdots+[a_m]x^m=a_0+a_1[x]+\cdots+a_m[x]^m$。这样，$L$ 中的任何一个元素都可以写成$[x]$的 K 上多项式的形式。因为任何一个既包含 K 又包含$[x]$的域必定包含这些 K 上的$[x]$的多项式，因此 L 是在 K 添加$[x]$得到的单扩域。如果写 $f(x)=b_0+b_1x+\cdots+b_nx^n$，那么 $f([x])=b_0+b_1[x]+\cdots+b_n[x]^n=[b_0+b_1x+\cdots+b_nx^n]=[f]=0$，所以$[x]$是 $f(x)$的一个根，而且 L 是 K 的单扩域。定理证毕。

例 2.3.2　设 $f(x)=x^2+x+2\in F_3[x]$，构造模$(f(x))$的剩余类环 $L=F_3[x]/(f(x))$，则 L 也(可看作)是 F_3 的扩域，且 $\theta=[x]=x+(f)$是 f 的一个根，$2\theta+2$ 是其另一个根。所以 $L=F_3(\theta)=\{0,1,2,\theta,\theta+1,\theta+2,2\theta,2\theta+1,2\theta+2\}$。容易验证，添加 $2\theta+2$ 到 F_3 上就会得到同一个域。

定义 2.3.6　设 E 是一域，$E[x]$是 E 上的多项式环。如果 $E[x]$中的每一个多项式在 $E[x]$中都可以分解成一次因式的乘积，则称 E 为代数闭域。

显然，代数闭域不再有真正的代数扩域。

定义 2.3.7　域 F 的一个扩域 E 叫做 $F[x]$中 n 次多项式 $f(x)$在 F 上的一个分裂域，如果 $f(x)$在 E 上可以分解成一次因式的乘积，而在任何一个 E 的真子域 K $(F\subset K\subset E)$上，$f(x)$都不能分解成一次因式的乘积。也就是说，E 是包含 $f(x)$的所有根的 F 的最小扩域，或者说 E 是一个使得 $f(x)$能够分解成一次因式之积的 F 的最小扩域。

显然有以下定理。

定理 2.3.8　令 E 是域 F 上多项式 $f(x)$的一个分裂域，

$$f(x)=a_n(x-\alpha_1)(x-\alpha_2)\cdots(x-\alpha_n),\quad \alpha_i\in E$$

那么，$E=F(\alpha_1,\cdots,\alpha_n)$。

定理 2.3.9　给了域 K 上的一多项式 $f(x)$。那么 $f(x)$在 K 上的任何两个分裂域是同构的。而且其同构映射可保持 F 中的元素不动，但把 $f(x)$的一个根映射成另外一个根。

证明：该定理的证明超出了本书的范围，有兴趣的读者可以参考文献[1]中的证明。

2.4　模与向量空间

2.4.1　向量空间

向量空间的理论，又称线性代数，是高等数学中的重要内容。在这一节将介绍向量空间理论。

定义 2.4.1　设 K 是一域，一个 K 上的向量空间 V 是一个加法群与一个称为数乘的运算：$K\times V\mapsto V$，对任意的 $\lambda,\mu\in K$ 和 $v,w\in V$ 满足：

(1) $\lambda\cdot(v+w)=\lambda\cdot v+\lambda\cdot w$；

(2) $(\lambda+\mu)\cdot v=\lambda\cdot v+\mu\cdot v$；

(3) $(\lambda\cdot\mu)\cdot v=\lambda\cdot(\mu\cdot v)$;

(4) $1\cdot v=v$。

我们把一个向量空间 V 中的元素称为向量，而把域 K 的元素称为数。

例 2.4.1 (1) 设 K 是一域，$V=\{0\}$是一平凡的 Abel 群，对任意 $\lambda\in K$，定义数乘运算 $\lambda\cdot 0=0$，则 V 是一 K 向量空间。

(2) K 是一域，$1\leqslant n\in\mathbb{N}$。在 K^n 中定义加法运算如下：

$$(v_1,\cdots,v_n)+(w_1,\cdots,w_n)=(v_1+w_1,\cdots,v_n+w_n)$$

则 K^n 构成一加法群，零元素为$(0,\cdots,0)$。如果进一步定义 $K\times K^n\to K^n$ 的数乘运算：

$$\lambda\cdot(v_1,\cdots,v_n)=(\lambda v_1,\cdots,\lambda v_n)$$

则 K^n 构成一 K 向量空间。我们把这个向量空间记为 $V_n(K)$。

(3) 设 F 是一域，K 是 F 的一子域。如果定义 $K\times F\to F$ 数乘运算为 F 中的乘法运算，则 F 称为一个 K 向量空间。

引理 2.4.1 设 V 是一 K 向量空间，$v\in V$，$\lambda\in K$。那么

(1) $0\cdot v=0$，$\lambda\cdot 0=0$；

(2) $(-1)\cdot v=-v$。

证明：(1) 注意到 $0\cdot v=(0+0)\cdot v=0\cdot v+0\cdot v$，所以 $0\cdot v=0$。同时，由于 $\lambda\cdot 0=\lambda\cdot(0+0)=\lambda\cdot 0+\lambda\cdot 0$，因此 $0=\lambda\cdot 0$。

(2) 考查方程：

$$0=0\cdot v=(1+(-1))\cdot v=1\cdot v+(-1)\cdot v=v+(-1)\cdot v$$

所以 $-v=(-1)\cdot v$。

定义 2.4.2 设 V 是一个 K 向量空间。我们称 V 的一个非空子集 U 是一个子空间，如果在 U 中加法运算和数乘运算都是封闭的，也就是说，任给 $v,w\in U$，$\lambda\in K$，都有 $v+w\in U$，$\lambda\cdot v\in U$。

从子空间的定义可看出，对任意的 $v\in U$，$0=0\cdot v\in U$，$-v=(-1)\cdot v\in U$，因此 U 实际上也是 V 的一个子群。

定义 2.4.3 设 V、W 是两个 K 向量空间。一个映射 ϕ: $V\to W$ 称为 K 向量空间的一个线性映射（或同态），如果对任意的 $u,v\in V$，$\lambda\in K$，有

$$\phi(u+v)=\phi(u)+\phi(v)$$
$$\phi(\lambda\cdot v)=\lambda\cdot\phi(v)$$

进一步，如果 ϕ 还是一个双射，则称 ϕ 是一个同构映射。

容易证明以下定理成立。

定理 2.4.1 设 ϕ: $V\to W$ 是一个 K 向量空间的线性映射，那么 $\phi(V)$是 W 的一个子空间，$\ker(\phi)=\phi^{-1}(0)$是 W 的一个子空间。

设 $v_1,v_2,\cdots,v_n$ 是 K 向量空间 V 中两两不同的元素，把以下形式的和式：

$$\sum_{i=1}^{n}\lambda_i\cdot v_i\quad \lambda_i\in K,\quad 1\leqslant i\leqslant n$$

称为向量 v_i 的线性组合，系数为 λ_i。为了方便起见，有时也把零个元素的线性组合

定义为 0。

定义 2.4.4　设 V 是一个 K 向量空间，B 是 V 的一个非空子集。

(1) 我们称 B 是线性无关的，如果对任何的 $n\in\mathbb{N}^+$，n 个互不相同的元素 $v_1,\cdots,v_n\in B$，都不存在不全为 0 的 $\lambda_1,\cdots,\lambda_n\in K$，使得 $\sum_{i=1}^{n}\lambda_i\cdot v_i=0$。

一个集合，如果不是线性无关的，则称其为线性相关的。

(2) 我们称 B 是 V 的一个生成元系，如果对任意的 $v\in V$，都存在 $n\in\mathbb{N}^+$，$v_1,\cdots,v_n\in B$，以及 $\lambda_1,\cdots,\lambda_n\in K$，使得

$$v=\sum_{i=1}^{n}\lambda_i\cdot v_i$$

(3) 我们称 B 是 V 的一组基，如果 B 是 V 的一个线性无关的生成元系。

容易看出，线性无关集的子集还是线性无关的，任何包含生成元系的集合也是生成元系。

例 2.4.2　设 K 是一域，$R=K[x_1,\cdots,x_n]$是 K 上的多变元多项式环，则 R 可以看成 K 向量空间 V。其中，数乘运算就是常数乘多项式的运算。V 中的线性组合就是多项式常数倍数之和，所有单项式(包括零次单项式)就构成了 V 的一组生成元系。

定理 2.4.2　设 V 是一个 K 向量空间，并有一个有限的生成元系 C，那么 V 一定有个基 $B\subset C$。

证明：令

$$N=\{|B|\mid B\subseteq C,\text{且 } B \text{ 是 } V \text{ 的有限生成元系}\}\subseteq\mathbb{N}$$

则 N 是一个非空集，因此有一个极小元 $n_0\in\mathbb{N}$。假设 $B_0\subseteq C$ 是 V 的一个有限生成元系，满足 $|B_0|=n_0$，则 B_0 中的元素是线性无关的，否则 B 中一定有一元素 v_0 可以写成 B_0 中其他元素的线性组合，这样 $B_0\backslash\{v_0\}$就成为一个更小的生成元系，这与 n_0 的最小性相矛盾。因此 B_0 中的元素是线性无关的，从而是 V 的一组基。

定理 2.4.3　设 V 是一个 K 向量空间，并假定有一个有限基 B，那么 V 中的每个线性无关集最多只有$|B|$个元素，而且 V 的任何一个生成元系都至少有$|B|$个元素，因此 V 的每组基正好有$|B|$个元素。

上述定理中的向量空间 V 称为有限维向量空间，维数为$|B|$，记做 $\dim_K(V)$，如果 V 不是有限维的，则称它是无穷维的，记为 $\dim_K(V)=\infty$。

推论 2.4.1　设 V 是一个有限维 K 向量空间，B 是 V 的一个有限集合，且$|B|=\dim_K(V)$，那么，B 是线性无关的当且仅当它是 V 的一个生成元系，也当且仅当它是 V 的一组基。

定理 2.4.4　同一域 K 上的两个同构的有限维向量空间一定有相同的维数。

证明：设 V 和 V'是 K 上的两个同构的向量空间，$\dim_K(V)=n$，$\dim_K(V')=m$，并假定

$$\sigma: V\rightarrow V'$$

是 V 到 V'的一个同构。那么 V 有一组基，它由 n 个向量 $e_1,\cdots,e_n$ 给成。下面来证

明 $\sigma(e_1),\cdots,\sigma(e_n)$在 K 上线性无关。设有线性关系

$$c_1\sigma(e_1)+\cdots+c_n\sigma(e_n)=0,\quad c_i\in K$$

因 σ 是同构,所以

$$\sigma(c_1e_1+\cdots+c_ne_n)=c_1\sigma(e_1)+\cdots+c_n\sigma(e_n)=0$$

还是因为 σ 是一同构,σ 是一一映射,所以

$$c_1e_1+\cdots+c_ne_n=0$$

这就证明了 $\sigma(e_1),\cdots,\sigma(e_n)$在 K 上线性无关,因此 $n\leqslant m$。同理,可证 $m\leqslant n$。所以 $m=n$。

定理 2.4.5 设 V 是域 K 上的一个 n 维向量空间,那么 V 一定和 $V_n(K)$同构。

证明:设 $e_1,e_2,\cdots,e_n$ 是 V 的一组基,那么 V 中任一个向量 v 都可以唯一地表示成 $e_1,e_2,\cdots,e_n$ 的线性组合

$$v=v_1e_1+v_2e_2+\cdots+v_ne_n,\quad v_i\in K$$

从而 v 唯一地确定了 $V_n(K)$中的一个向量$(v_1,v_2,\cdots,v_n)$。反过来,v 又由$(v_1,v_2,\cdots,v_n)$唯一确定。这样就定义了一个从 V 到 $V_n(K)$的一一映射:

$$\sigma:v\mapsto(v_1,v_2,\cdots,v_n)$$

再设 $w\in V$,那么 w 也可以唯一地表示成 $e_1,e_2,\cdots,e_n$ 的线性组合

$$w=w_1e_1+w_2e_2+\cdots+w_ne_n$$

于是

$$v+w=(v_1+w_1)e_1+(v_2+w_2)e_2+\cdots+(v_n+w_n)e_n$$

因此

$$\sigma(v+w)=(v_1+w_1,v_2+w_2,\cdots,v_n+w_n)=\sigma(v)+\sigma(w)$$

设 $c\in K$,那么

$$c\cdot v=c(v_1e_1+v_2e_2+\cdots+v_ne_n)=(cv_1)e_1+(cv_2)e_2+\cdots+(cv_n)e_n$$

因此

$$\sigma(c\cdot v)=(cv_1,cv_2,\cdots,cv_n)=c(v_1,v_2,\cdots,v_n)=c\sigma(v)$$

这证明了 σ 是个同构。

2.4.2 模

前一小节讲了域上的向量空间,这一小节将介绍模的概念。模实际上是向量空间的一种推广,可以看成是一种环上的向量空间。首先看下面的定义。

定义 2.4.5 设 R 是环。加法群 M 被称为一个 R 模,是指对于一个称为数乘的运算$\circ$:$R\times M\mapsto M$,对任意的 $\alpha,\beta\in R$ 和 $a,b\in M$ 满足:

(1) $\alpha\circ(a+b)=\alpha\circ a+\alpha\circ b$;

(2) $(\alpha+\beta)\circ a=\alpha\circ a+\beta\circ a$;

(3) $(\alpha\dot{\beta})\circ a=\alpha\circ(\beta\circ a)$;

(4) $1\circ a=a$。

为方便起见,在不至于引起混淆的情况下,用 ab 表示环中的乘法运算 $a\dot{b}$,αa 表

示数乘运算 $\alpha\circ a$。

例 2.4.3　设 R 是一环。

(1) 设 I 是 R 的一个理想，则 I 相对于环 R 的加法运算和乘法运算形成一个 R 模。特别地，R 本身可以看成一个 R 模，零理想 $\{0\}$ 也形成一个 R 模。

(2) 设 $M=\{0\}$ 是一个平凡的加法群，对任意的 $\alpha\in R$，定义 $\alpha\circ 0=0$，则 M 是一个(平凡的) R 模。

(3) 设 $M=R^n$ 是 R 的一个有限直积。如果在 M 中定义加法运算：$(\alpha_1,\cdots,\alpha_n)+(\beta_1,\cdots,\beta_n)=(\alpha_1+\beta_1,\cdots,\alpha_n+\beta_n)$ 和数乘运算：$\alpha\circ(\alpha_1,\cdots,\alpha_n)$，则 M 是一个 R 模，这个模称为秩为 n 的自由 R 模。

(4) 每个环 R 上的多项式环 $R[X_1,\cdots,X_n]$ 都是一个 R 模，如果定义数乘运算为多项式环中的常数与多项式的乘法。

定义 2.4.6　设 M 是一 R 模，N 是 M 的一加法子群。如果 N 在数乘运算下是封闭的，则称 N 是 M 的子模。

定义 2.4.7　两个 R 模 M 和 M' 之间的一个映射 $\phi: M\mapsto M'$，如果对任何的 a，$b\in M$ 和 $\alpha\in R$ 都满足：

$$\phi(a+b)=\phi(a)+\phi(b)$$
$$\phi(\alpha\circ a)=\alpha\circ\phi(a)$$

则称 ϕ 是一个 R 模同态映射，而 0 的所有原像称为同态映射的核，记为 $\ker(\phi)=\phi^{-1}(0)=\{x\in M\mid\phi(x)=0\}$。从一个 R 模 M 到自身的同态映射称为 M 的自同态。

设 B 是 M 的一个子集，把包含 B 的最小的子模 N 称为由 B 在 M 中生成的子模。显然，N 包含了所有 B 中元素的线性组合：

$$N=\left\{\sum_{i=1}^{n}\alpha_i a_i\ \middle|\ \alpha_i\in R, a_i\in B\right\}$$

如果存在一个有限的集合 B 正好生成了 M，则称 M 是有限生成的 R 模。如果一个模的任何一个子模都是有限生成的，则称其为诺特(Noether)模。只有一个元素 x 生成的模称为循环模，可记为 $Rx=\{ax\mid a\in R\}$。

设 M 是一个 R 模，N 是其子模，则商群 M/N 在数乘运算：

$$\alpha(a+N)=\alpha a+N$$

下也构成一 R 模，称为 M 相对于子模 N 的分式模。

定理 2.4.6(同态定理)　设 $\psi: M\mapsto M'$ 是一个模同态，$N\subset\ker\psi$ 是 M 的一个子模，$\phi: M\mapsto M/N$，$\phi(a)=a+N$ 是 M 到 M/N 的典范同态，那么映射 $\eta: M/N\mapsto M'$，$\eta(a+N)=\psi(a)$ 是良定义的，且是同态映射，满足 $\eta\circ\phi=\psi$，而且 η 是满射当且仅当 ψ 是满射，η 是单映射当且仅当 $N=\ker\psi$。

设 $x\in M$，定义 $ann(x)=\{d\in R\mid dx=0\}$，则 $ann(x)$ 是 R 的一子环，利用上面的同态定理，容易证明

$$Rx\cong R/ann(x)$$

定理 2.4.7　设 M 是一个由 $(b_1,\cdots,b_n)$ 生成的 R 模。那么映射：

$$\begin{array}{cccc} \phi: & R^n & \mapsto & M \\ & (a_1,\cdots,a_n) & & \sum_{i=1}^{n} a_i b_i \end{array}$$

是一个满同态，且 M 同构于 $R^n/\ker(\phi)$。

定理 2.4.8 (1) 设 M 和 M' 是 R 模，$\phi: M \mapsto M'$ 是一个满同态，则如果 M 是诺特模，那么 M' 也是诺特模。

(2) 设 M 是一个诺特 R 模，N 是其一子模，则 N 和 M/N 也都是诺特 R 模。

(3) 设 $M=R^n$ 是诺特环 R 上的秩为 n 的自由 R 模，那么 M 也是诺特 R 模。

(4) 如果 M 是一个诺特环上有限生成的 R 模，那么 M 一定是诺特 R 模。

证明：(1) 假设 N' 是 M' 的任意一个子模，则 $N=\phi^{-1}(N')$ 是 M 的一个子模，因此是有限生成的，从而 $N'=\phi(\phi^{-1}(N'))$ 也是有限生成的，事实上，N 的生成元在 ϕ 下的像正好生成 N'。因此 M' 是诺特模。

(2) 假设 N' 是 N 的一个子模，则 N' 也是 M 的子模，因此是有限生成的，从而推得 N 是诺特模。现令 ϕ 是 M 到 M/N 的典范映射，则 ϕ 是一个满射，从而根据(1)，M/N 也是诺特模。

(3) 用数学归纳法。$n=1$ 时定理是显然的，现假设 $n>1$，R^{n-1} 是诺特模，并设 N 是 R^n 的一个子模。下面证明 N 是有限生成的。令 π 是下面的投影映射：

$$\begin{array}{cccc} \phi: & R^n & \mapsto & R^{n-1} \\ & (a_1,\cdots,a_n) & & (a_1,\cdots,a_{n-1}) \end{array}$$

并令

$$I = \{r \in R \mid (0,\cdots,0,r) \in N\}$$

容易验证 $\pi(N)$ 是 R^{n-1} 的一个子模，I 是 R 的一个理想。因此根据定理假设和归纳假设，$\pi(N)$ 和 I 分别有限生成元组 B 和 C。设 D 是 N 中的一有限集使得 $\pi(D)=B$，并令

$$E = \{(0,\cdots,0,r) \mid r \in C\}$$

则 E 也是 N 的子集。事实上 $D\cup E$ 就是 N 的一个生成元组。为了证明这点，任取 $a\in N$，由于 $\pi(a)\in\pi(N)$，因此一定存在 $\alpha_1,\cdots,\alpha_k\in R$ 和 $b_1,\cdots,b_k\in B$，使得

$$\pi(a) = \sum_{i=1}^{k} \alpha_i b_i$$

根据 D 的选取，又一定存在 $d_1,\cdots,d_k\in D$，使得 $\pi(d_i)=b_i$，$1\leqslant i\leqslant k$。取

$$b = a - \sum_{i=1}^{k} \alpha_i d_i$$

则 $b\in N$ 且 $\pi(b)=0$，因此 $b=(0,\cdots,0,r)$ 对某 $r\in R$ 成立。容易看出 $r\in I$，因此存在 $\beta_1,\cdots,\beta_l$，使得

$$r = \sum_{i=1}^{k} \beta_i c_i$$

令 $e_i=(0,\cdots,0,c_i)\in E$，则

$$b=(0,\cdots,0,r)=\sum_{i=1}^{l}\beta_i e_i$$

综上，有

$$a=b+\sum_{i=1}^{k}\alpha_i d_i=\sum_{i=1}^{l}\beta_i e_i+\sum_{i=1}^{k}\alpha_i d_i$$

因此 $D\cup E$ 是 N 的生成元组。

(4) 设 $\{b_1,\cdots,b_n\}$ 是一个有限生成元组。则根据定理 2.4.8，映射

$$\phi:\quad R^n \mapsto M$$
$$(a_1,\cdots,a_n)\quad \sum_{i=1}^{n}a_i b_i$$

是一个满同态，根据定理的(1)和(3)，结论即得。

2.5　有限域与 Galois 环

在这一节将要介绍有限域的一些基本性质。有限域作为一种只含有有限多个元素的特殊的域，有着许多其他域所没有的特殊性质，比如说每一个有限域中元素的个数一定是某一素数的幂，而且对任一素数幂，也一定存在相应的有限域；再比如说，任何两个元素个数相同的有限域一定同构，从而可以把它们等同起来等。这一节首先介绍有限域的一些特征性质，然后介绍有限域中元素的迹、范数，最后介绍一种和有限域非常相像的环——Galois 环。

2.5.1　有限域及其性质

在 2.2 节曾经碰到过一类由整数环中的剩余类构成的有限域(例 2.2.7)，即对任一的素数 p，模 p 的剩余类环 $Z_p=Z/(p)$ 形成一个含有 p 个元素的有限域 F_p。这是一类非常重要的有限域，因为任何一个特征为 p 的域一定包含一个和 F_p 同构的子域(参看定理 2.3.1)，因此可以看成是 F_p 的一个扩域。这一结果正是有限域的分类和构造的基础。

定义 2.5.1　设 F 是域。如果存在正整数 n 使得对任何 $f\in F$，$n\cdot f=0$，但对任何小于 n 的正整数 n'，$n'\cdot f\neq 0$，则称 n 为域 F 的特征，否则则称 F 的特征为 0。域 F 的特征记为 $\mathrm{ch}(F)$。

定理 2.5.1　有限域的特征是一素数。

证明：此处略，作为练习。

定理 2.5.2　在特征为 p 的有限域中，$(a\pm b)^{p^n}=a^{p^n}\pm b^{p^n}$。

证明：下面只证明 $(a+b)^{p^n}=a^{p^n}+b^{p^n}$，而把减号的情形留作习题。我们对 n 用归纳法证明。

当 $n=1$ 时，

$$(a+b)^p=a^p+\sum_{k=1}^{p-1}\binom{p}{k}a^{p-k}b^k+b^p$$

由于 p 是素数，因此对于 $1\leqslant k<p$，$(p,k!(p-k)!)=1$，所以 $p\left|\dfrac{p\cdot(p-1)!}{k!(p-k)!}\right.$，即 $p\left|\dbinom{p}{k}\right.$。因此 $(a+b)^p=a^p+b^p$。

假设 $n=k$ 时定理成立，即 $(a+b)^{p^k}=a^{p^k}+b^{p^k}$，那么当 $n=k+1$ 时，有

$$\begin{aligned}(a+b)^{p^{k+1}}&=((a+b)^{p^k})^p=(a^{p^k}+b^{p^k})^p=(a^{p^k})^p+(b^{p^k})^p\\&=a^{p^{k+1}}+b^{p^{k+1}}\end{aligned}$$

定理得证。

定理 2.5.3 假设 F 是一个有限域，K 为其子域。如果 K 有 q 个元素，则 F 有 q^m 个元素，其中 $m=[F:K]$。

证明：F 可以看作 K 上的向量空间。如果 $m=[F:K]$，则 F 存在由 m 个元素组成的基底 $b_1,b_2,\cdots,b_m$，且 F 中的每一个元素都可唯一地表示成 $\sum a_ib_i,a_i\in K$，所以 F 有 q^m 个元素。

定理 2.5.4 假设 F 是一个有限域，p 是 F 的特征。则 F 有 p^n 个元素，其中 n 是 F 关于其素域的扩张次数。

证明：由于 F 的特征为 p，则 F 的素域同构于 F_p。因此含有 p 个元素。由定理 2.5.3 知F 有 p^n 个元素。

定理 2.5.5 假设 F 是具有 q 个元素的有限域，则 $\forall a\in F,a^q=a$。

证明：首先 $a^q=a$ 对 $a=0$ 成立。F 中所有非零元素组成一个 $q-1$ 阶有限群，所以 $a^{q-1}=1$，从而 $a^q=a$ 对所有 $a\neq 0$ 成立。这样就证明了 $a^q=a$ 对所有 $a\in F$ 成立。

定理 2.5.6 如果 F 是有 q 个元素的有限域，K 为一子域，则 $K[x]$中的多项式 x^q-x 在 $F[x]$中可分解为：$x^q-x=\prod_{a\in F}(x-a)$且 F 是 K 上多项式 x^q-x 的分裂域。

证明：我们知道 x^q-x 在 F 中至多含有 q 个根，而根据定理 2.5.5，F 中的 q 个元素都是这个多项式的根。所以 x^q-x 在 F 中是分裂的，而且不能在任何更小的域中分裂。

定理 2.5.7（存在，唯一性定理） 对任何素数 p 和正整数 n，存在一个有限域含有 p^n 个元素。且任何具有 $q=p^n$ 个元素的有限域同构于 x^q-x 在 F_p上的分裂域。

证明：（存在性）对 $q=p^n$，考虑 F_p上的多项式 x^q-x。假设 F 是 F_p上 x^q-x 的分裂域。我们知道 x^q-x 有 q 个不同的根，令 S 是 F 中多项式 x^q-x 的所有根组成的集合。则：

1）$0,1\in S$；

2）$\forall a,b\in S$，因为$(a-b)^q=a^q-b^q=a-b$，所以 $a-b\in S$；

3）$\forall a,b\in S$，由于$(ab^{-1})^q=a^q\cdot b^{-q}=ab^{-1}$，因此 $ab^{-1}\in S$。

综上所述，所以 S 为 F 的一子域。另一方面 x^q-x 在 S 中分裂，所以 $S=F$，因此 F 具有 q 个元素。

（唯一性）假设 F 是具有 $q=p^n$ 个元素的有限域，则 F 的特征为 p 且以 F_p为其子域。所以 F 是 F_p上多项式 x^q-x 的分裂域。根据定理 2.3.9，多项式的分裂域（在

同构意义下)是唯一的,所以具有 q 个元素的有限域唯一,且都同构于 x^q-x 在 F_p 上的分裂域。

定理 2.5.8(子域准则)　假设 F_q 是一个具有 $q=p^n$ 个元素的有限域,则 F_q 的每一个子域含有 p^m 个元素,且 $m|n$。反之,对 n 的任一正因子 m,也存在唯一的 F_q 的子域含有 p^m 个元素。

证明:假设 K 是 F_q 的子域。显然对某一 $m\in N^+$,有 $|K|=p^m$。由定理 2.5.3,知 q 是 $|K|$ 的某一幂次,所以 $(p^m)^r=p^n$,因此 $m|n$。反之,假设 $m|n$,则 $p^m-1|p^n-1$,所以 $x^{p^m-1}-1|x^{p^n-1}-1$,因此 $x^{p^m}-x|x^{p^n}-x$。从而 $x^{p^m}-x$ 的分裂域为 F_q 的子域且此子域含有 p^m 个元素。假设 F_q 有两个不同的含有 p^m 个元素的子域,那么这两个子域中元素都是 $x^{p^m}-x$ 的根,因此这两个子域一定相同。

例 2.5.1　有限域 $F_{2^{12}}$ 的子域完全由 12 的因子决定。12 有 6 个因子即 1、2、3、4、6、12,它们所对应的子域分别是 F_2、F_{2^2}、F_{2^3}、F_{2^4}、F_{2^6}、$F_{2^{12}}$。

定理 2.5.9　对每一个有限域 F_q,其乘法群 F_q^* 是 $q-1$ 阶的循环群。

证明:假设 $q\geqslant3$ 且 $h=q-1=p_1{}^{r_1}p_2{}^{r_2}\cdots p_m{}^{r_m}$,其中 $r_i\geqslant1$。现构造阶为 $p_i{}^{r_i}$ 的元素 b_i 如下:对每一个 i,多项式 $x^{h/p_i}-1$ 最多只有 h/p_i 个根在 F_q 中,所以 F_q 中至少有一个 a_i 不是 $x^{h/p_i}-1$ 的根,即 $a_i{}^{h/p_i}\neq1$. 令 $b_i=a_i{}^{h/p_i^{r_i}}$,则 $b_i{}^{p_i^{r_i}}=a_i^h=1$,所以 b_i 的阶整除 $p_i^{r_i}$。但 $b_i^{p_i^{r_i-1}}=a_i^{h/p_i}\neq1$,所以 b_i 的阶为 $p_i^{r_i}$。下面证明 $b=b_1b_2\cdots b_m$ 的阶为 h。否则的话,b 的阶至少为某一 h/p_i 的因子。不妨假设 b 的阶为 h/p_1 的因子,那么 $1=b^{h/p_1}=b_1{}^{h/p_1}\cdots b_m^{h/p_1}=b_1^{h/p_1}$,所以 $p_1^{r_1}|h/p_1$。而这是不可能的,所以 b 的阶为 h。

定义 2.5.2　F_q^* 中的生成元称为 F_q 的本原元。

显然 F_q 中有 $\phi(q-1)$ 个本原元。

定理 2.5.10　设 F_q 是一个有限域,F_r 是 F_q 的一个有限扩域,则 F_r 为 F_q 的一个单扩张,且任一 F_r 的本原元都是 F_r 在 F_q 上的定义元。

证明:假设 ξ 为 F_r 的本原元。显然 $F_q(\xi)\subset F_r$。同时由于 ξ 是 F_r 的本原元,所以有 $F_r\subset F_q(\xi)$,因此 $F_r=F_q(\xi)$。

定理 2.5.11　对任意的有限域 F_q 和正整数 n,一定存在 F_q 上的 n 次不可约多项式。

证明:根据定理 2.5.7,存在有限域 F_{q^n}。显然,F_{q^n} 是 F_q 的 n 次扩张,即 $[F_{q^n}:F_q]=n$。根据定理 2.5.10,设 $\xi\in F_{q^n}$ 是 F_{q^n} 的本原元,则 $F_{q^n}=F_q(\xi)$,从而知 ξ 的极小多项式的次数 n,而这是 F_q 上的一个不可约多项式,因此定理得证。

定理 2.5.12　设 $f(x)\in F_q[x]$ 是 m 次不可约多项式。则 $f(x)|x^{q^n}-x\Leftrightarrow m|n$。

证明:假设 $f(x)|x^{q^n}-x$。α 是 $f(x)$ 在某一个分裂域中的根,则 $\alpha^{q^n}=\alpha$,所以 $\alpha\in F_{q^n}$,因此 $F_q(\alpha)\subset F_{q^n}$。但 $[F_q(\alpha):F_q]=m$,$[F_{q^n}:F_q]=n$,所以 $m|n$。反之,如果 $m|n$,则 F_{q^m} 可以看作 F_{q^n} 的子域。如果 α 是 $f(x)$ 在某一个分裂域中的一个根,则 $[F_q(\alpha):F_q]=m$,所以 $F_q(\alpha)=F_{q^m}$,因此 $\alpha\in F_{q^n}$,从而 $\alpha^{q^n}=\alpha$,即 α 是 $x^{q^n}-x$ 的根。所以 $f(x)|x^{q^n}-x$。

定理 2.5.13 设 $f(x)$是 $F_q[x]$中次数为 m 的不可约多项式，则 $f(x)$有根 α 在 F_{q^m} 中。进一步，$f(x)$的所有根正好为 F_{q^m} 中以下 m 个元素：$\alpha,\alpha^q,\alpha^{q^2},\cdots,\alpha^{q^{m-1}}$。

证明： 假设 α 是 $f(x)$在某一分裂域中的根，则$[F_q(\alpha):F_q]=m$，所以 $F_q(\alpha)=F_{q^m}$，$\alpha\in F_{q^m}$。下面证明如果 $\beta\in F_{q^m}$ 是 $f(x)$的一个根，则 β^q也是 $f(x)$的根。写 $f(x)=a_m x^m+\cdots+a_1x+a_0$，$a_i\in F_q$，那么根据定理 2.5.2 和定理 2.5.5，有

$$f(\beta^q)=a_m(\beta^q)^m+\cdots+a_1\beta^q+a_0=(a_m\beta^m+\cdots+a_1\beta+a_0)^q=0$$

所以 $\alpha,\alpha^q,\cdots,\alpha^{q^{m-1}}$ 都是 $f(x)$的根。下面证明这些元素互不相同。假设 $\alpha^{q^j}=\alpha^{q^k}$，$0\leqslant j<k\leqslant m-1$，则$(\alpha^{q^j})^{q^{m-k}}=(\alpha^{q^k})^{q^{m-k}}$，所以 $\alpha^{q^{m-k+j}}=\alpha^{q^m}=\alpha$，从而 $f(x)\mid x^{q^{m-k+j}}-x$。由定理 2.5.12，得 $m\mid m-k+j$，这与 $k>j$ 矛盾。

在这里，对定理 2.5.13 中出现的元素引入一个新的概念，而不论 $\alpha\in F_{q^m}$ 是否是 F_q 上某一 m 次不可约多项式的根。

定义 2.5.3 假设 F_{q^m} 是 F_q 的扩域，$\alpha\in F_{q^m}$，则 $\alpha,\alpha^q,\cdots,\alpha^{q^{m-1}}$ 称为 α 相对于 F_q 的共轭元。

定理 2.5.14 $\alpha\in F_q^*$ 相对于任一子域的共轭元在 F_q^* 中有相同的阶。

证明： 因为$\circ(\alpha)\mid q-1$，且对任意的正整数 i，$(q^i,q-1)=1$，所以根据定理 2.1.2，α^{q^i} 的阶都为$\circ(\alpha)$。

推论 2.5.1 如果 α 是 F_q 的本原元，则 α 相对于任一子域的共轭元也是本原元。

例 2.5.2 $f(x)=x^4+x+1\in F_2[x]$，$\alpha\in F_{16}$，则 α 相对于 F_2 的共轭元为 α,α^2，$\alpha^4=\alpha+1$，$\alpha^8=\alpha^2+1$。但 α 相对于 F_4 的共轭元为 $\alpha,\alpha^4=\alpha+1$。

2.5.2 元素的迹

定义 2.5.4 对 $\alpha\in F=F_{q^m}$，$K=F_q$，定义 α 的迹 $Tr_{F/K}(\alpha)$如下：

$$Tr_{F/K}(\alpha)=\alpha+\alpha^q+\cdots+\alpha^{q^{m-1}}$$

如果 K 是 F 的素域，则 $Tr_{F/K}(\alpha)$称为 α 的绝对迹，记做 $Tr_F(\alpha)$。

换句话说，α 相对于 K 的迹就是所有 α 的相对于 K 的共轭元之和。下面就来说明 $Tr_{F/K}(\alpha)\in K$。

定理 2.5.15 设 $\alpha\in F=F_{q^m}$，$K=F_q$，则 $Tr_{F/K}(\alpha)\in K$。

证明： 不难验证$(Tr_{F/K}(\alpha))^q=Tr_{F/K}(\alpha)$，所以 $Tr_{F/K}(\alpha)$是 x^q-x 的根，因此 $Tr_{F/K}(\alpha)\in F_q=K$。

定理 2.5.16 设 $K=F_q$，$F=F_{q^m}$，则迹函数 $Tr_{F/K}$满足：

1） $Tr_{F/K}(\alpha+\beta)=Tr_{F/K}(\alpha)+Tr_{F/K}(\beta)$；

2） $Tr_{F/K}(c\alpha)=cTr_{F/K}(\alpha)$，$c\in K$，$\alpha\in F$；

3） $Tr_{F/K}$是 F 到 K 上的线性变换，这里把 F 和 K 都看作 K 上的向量空间；

4） $Tr_{F/K}(a)=ma$，$a\in K$；

5） $Tr_{F/K}(\alpha^q)=Tr_{F/K}(\alpha)$，$\forall\alpha\in F$。

证明： 1)和 2)保证了 $Tr_{F/K}$为 F 到 K 的一个线性变换，因此只要证 $Tr_{F/K}$是满的即可。首先证明 $\exists\alpha\in F$，使得 $Tr_{F/K}(\alpha)\neq 0$。因为 $Tr_{F/K}(\alpha)=0\Leftrightarrow\alpha$ 是 $x^{q^{m-1}}+\cdots+$

x^q+x 的根，而该多项式最多只有 q^{m-1} 个根，F 中却有 q^m 个元素，因此存在 $\alpha\in F$，使 $Tr_{F/K}(\alpha)\neq 0$。假设 $b=Tr_{F/K}(\alpha)\neq 0$，那么 $b\in K$，且对任给的 $a\in K$，$ab^{-1}\in K$，因此 $Tr_{F/K}(ab^{-1}\alpha)=(ab^{-1})Tr_{F/K}(\alpha)=(ab^{-1})b=a$，所以 $Tr_{F/K}$ 是 F 到 K 的一个满射。

从上面的定理可知，如果把 F 和 K 都看作 K 上的向量空间，迹映射是从 F 到 K 的一个线性变换。从下面的定理可以看到，迹映射不仅仅自己是一个线性变换，而且还可以从它导出其他所有的线性变换，从而使对线性变换的描述可以不依赖于基的选取。有时也把从 F 到 K 的线性变换称为 F/K 上的线性泛函(linear function)，或简称 F 上的线性泛函。F 上的所有线性泛函的全体在线性变换的加法和数乘运算下也构成 K 上的一个向量空间，这个向量空间称为 F 的对偶向量空间。

定理 2.5.17　设 F 是有限域 K 的有限扩张。把 F 和 K 都看作 K 上的向量空间，则 F 到 K 的所有线性变换正好是以下形式的映射 L_β：$L_\beta(\alpha)=Tr_{F/K}(\beta\alpha)$，其中 $\beta\in F$。进一步，如果 β 和 γ 是不同的元素，则 $L_\beta\neq L_\gamma$。

证明：不妨设 $K=F_q$，$F=F_{q^m}$。根据定理 2.5.16 的 3)，L_β 是线性变换是显然的。现在假设 $\beta\neq\gamma$。由于 $Tr_{F/K}$ 是 F 到 K 上的线性变换，所以 $\exists\alpha\in F$ 使得 $Tr_{F/K}((\beta-\gamma)\alpha)\neq 0$，因此 $L_\beta(\alpha)-L_\gamma(\alpha)=Tr_{F/K}((\beta-\gamma)\alpha)\neq 0$，从而 $L_\beta\neq L_\gamma$。这样 $\{L_\beta:\beta\in F\}$ 共给出 q^m 个线性变换。但由于 $F\to K$ 的任何一个线性变换都可以由该变换作用在某一组基上的像完全确定，而且这些像可以在 K 中任意取值，因此 $F\to K$ 的线性变换共有 q^m 个。所以 $\{L_\beta:\beta\in F\}$ 就是线性变换的全体。

从迹的定义可以看出，元素的迹和所考虑的子域有关。相对于域的包含关系，迹映射有下面的传递性。

定理 2.5.18(传递性)　假设 K 是一个有限域，F 是 K 的有限扩张，E 是 F 的有限扩张，那么 $\forall\alpha\in E$，$Tr_{E/K}(\alpha)=Tr_{F/K}(Tr_{E/F}(\alpha))$。

证明：设 $K=F_q$，$[F:K]=m$，$[E:F]=n$，则 $[E:K]=mn$，$F=F_{q^m}$，$E=F_{q^{mn}}$。$\forall\alpha\in E$，我们有 $Tr_{F/K}(Tr_{E/F}(\alpha))=\sum\limits_{i=0}^{m-1}(Tr_{E/F}(\alpha))^{q^i}=\sum\limits_{i=0}^{m-1}\left(\sum\limits_{j=0}^{n-1}\alpha^{q^{jm}}\right)^{q^i}=\sum\limits_{i=0}^{m-1}\sum\limits_{j=0}^{n-1}\alpha^{q^{jm+i}}=\sum\limits_{k=0}^{mn-1}\alpha^{q^k}=Tr_{E/K}(\alpha)$。

设 $K=F_q$ 是有限域，$F=F_{q^m}$ 是 K 的 m 次有限扩张。已经知道，F 可以看成是 K 上的 m 维向量空间，因此一定存在一组基，而 F 中的任一元素都可以唯一地表示成这组基的线性表达式。下面要讨论的问题来自于对基表达式中系数的计算。设 α_1，α_2，$\cdots$，α_m 是 F 在其子域 K 上的一组基，则对 $\forall\alpha\in F$，有唯一的表达式 $\alpha=c_1(\alpha)\alpha_1+\cdots+c_m(\alpha)\alpha_m$。易知 c_j：$\alpha\to c_j(\alpha)$ 是 $F\to K$ 的线形变换。根据定理 2.5.17，存在 β_j 使得 $c_j(\alpha)=Tr_{F/K}(\beta_j\alpha)$。显然，如果 $i\neq j$，$Tr_{F/K}(\alpha_i\beta_j)=0$，否则 $Tr_{F/K}(\alpha_i\beta_j)=1$。进一步，$\beta_1$，$\cdots$，$\beta_m$ 正好也组成 F 在 K 上的一组基。因为如果 $\sum d_i\beta_i=0$，则 $\sum d_i\alpha_j\beta_i=0\Rightarrow\sum d_iTr_{F/K}(\alpha_j\beta_i)=0\Rightarrow d_j=0$，$j=1,2,\cdots,m$。从而有以下定义。

定义 2.5.5　设 K 是一个有限域，F 是 K 的有限扩张。F 在 K 上的两组基 $\{\alpha_1,\alpha_2,\cdots,\alpha_m\}$ 和 $\{\beta_1,\beta_2,\cdots,\beta_m\}$ 是对偶的(互补的)。如果对 $1\leqslant i,j\leqslant m$ 有

$$Tr_{F/K}(\alpha_i\beta_j) = \begin{cases} 0 & i \neq j \\ 1 & i = j \end{cases}$$

上面的讨论说明了对任何给定的基底，一定存在一对偶基 $\beta_1,\beta_2,\cdots,\beta_m$。事实上这一对偶基是唯一决定的，这是因为 β_j 是由 c_j 唯一决定的。

例 2.5.3 设 $\alpha\in F_8$ 是不可约多项式 $x^3+x^2+1\in F_2[x]$的一个根，则$\{\alpha,\alpha^2,1+\alpha+\alpha^2\}$是 F_8 在 F_2 上的一组基，可以验证由它决定的唯一的一组对偶基仍是$\{\alpha,\alpha^2,1+\alpha+\alpha^2\}$，这样的基叫**自对偶基**。$\alpha^5\in F_8$ 可以表示为 $\alpha^5=c_1\alpha+c_2\alpha^2+c_3(1+\alpha+\alpha^2)$，其中 $c_1=Tr_{F_8}(\alpha\cdot\alpha^5)=0$，$c_2=Tr_{F_8}(\alpha^2\cdot\alpha^5)=1$，$c_3=Tr_{F_8}((1+\alpha+\alpha^2)\cdot\alpha^5)=1$。所以 $\alpha^5=\alpha^2+(1+\alpha+\alpha^2)$。

域 F 在 K 上可以有好多基底，但有两类是人们比较感兴趣的。一种是多项式基底 $1,\alpha,\alpha^2,\cdots,\alpha^{m-1}$，这里 α 一般取为本原元。另外一种就是：

定义 2.5.6 设 $K=F_q$，$F=F_{q^m}$，则 F 在 K 上形如$\{\alpha,\alpha^q,\cdots,\alpha^{q^{m-1}}\}$（$\alpha$ 为某一适当的元素）的基称为正规基。

例 2.5.4 在例 2.5.3 中，容易验证 $\alpha^4=1+\alpha+\alpha^2$，因此$\{\alpha,\alpha^2,1+\alpha+\alpha^2\}=\{\alpha,\alpha^2,\alpha^4\}$实际上也是一个正规基。

2.5.3 多项式的阶

我们都知道，多项式的次数是多项式的一个非常重要的指标。实际上，对于有限域上的非零多项式，还有另外一个指标也非常重要，那就是多项式的阶。多项式阶的定义主要基于以下事实：

定义 2.5.7 设 $f(x)\in F_q[x]$是一非零多项式。如果 $f(0)\neq 0$，则定义 $f(x)$的阶(order)为满足 $f(x)\mid x^e-1$ 的最小的正整数 e，并记做 $\mathrm{ord}(f)$；如果 $f(0)=0$，那么一定可以写成 $f(x)=x^h\cdot g(x)$，其中 $g(0)\neq 0$，这时就把 $f(x)$的阶定义为 $g(x)$的阶。

多项式 $f(x)$的阶，有时也称为多项式 $f(x)$的周期，并记做 $p(f)$，或称为多项式的指数。

定理 2.5.19 设 $f(x)\in F_q[x]$是一 m 次不可约多项式，$f(0)\neq 0$，则 $\mathrm{ord}(f)$等于 $f(x)$的任一根在乘法群 $F_{q^m}^*$ 中的阶。

证明：我们知道 F_{q^m} 是 $f(x)$的分裂域，而且 $f(x)$的所有根都有相同阶（因为所有的根都是共轭的）。设 $\alpha\in F_{q^m}^*$ 是 $f(x)$的根，则由 $f(x)$的不可约性可知，$\alpha^e=1\Leftrightarrow f(x)\mid x^e-1$，所以 α 的阶就是 $f(x)$的阶。

推论 2.5.2 设 $f(x)\in F_q[x]$是 F_q 上的 m 次不可约多项式，则 $\mathrm{ord}(f(x))\mid q^m-1$。

证明：如果 $f(x)=c\cdot x$，$c\in F_q^*$，则 $\mathrm{ord}(f(x))=1$，所以定理成立。不然就有 $\mathrm{ord}(f(x))$等于 $f(x)$的根在 $F_{q^m}^*$ 中的阶，所以应整除 q^m-1。

前面给出了多项式阶的定义，并讨论了不可约多项式阶的一些性质，但怎样去求一个一般多项式的阶呢？由于任何一个多项式都可以写成不可约多项式的乘积。因此只要有办法求不可约多项式幂次的阶和互素多项式乘积的阶，就可以求任意多项式的阶。请看下面的定理。

引理 2.5.1　设 c 是一正整数，$f(x)\in F_q[x]$，$f(0)\neq 0$，则 $f(x)\mid x^c-1$ 当且仅当 $\mathrm{ord}(f(x))\mid c$。

证明：$\Leftarrow$：假设 $e=\mathrm{ord}(f(x))$ 整除 c，则 $f(x)\mid x^e-1$ 且 $x^e-1\mid x^c-1$，所以 $f(x)\mid x^c-1$。

$\Rightarrow$：假设 $f(x)\mid x^c-1$，则 $c\geqslant e$。写 $c=e\cdot m+r$，其中 m,r 都是整数，$0\leqslant r<e$。由于 $x^c-1=(x^{me}-1)x^r+(x^r-1)$，$f(x)\mid x^c-1$ 且 $f(x)\mid x^e-1$，所以 $f(x)\mid x^r-1$。但 $r<e$，所以只有 $r=0$，因此 $c=m\cdot e$，即 $e\mid c$。

引理 2.5.2　如果 e_1 和 e_2 是正整数，则 $x^{e_1}-1$ 和 $x^{e_2}-1$ 在 $F_q[x]$中的最大公因子是 x^d-1，其中 $d=(e_1,e_2)$。

证明：设 $f(x)$是 $x^{e_1}-1$ 和 $x^{e_2}-1$ 的最大公因子，则由于 x^d-1 是 $x^{e_1}-1$ 和 $x^{e_2}-1$ 的公因子，所以 $x^d-1\mid f(x)$。另一方面由于 $f(x)$是 $x^{e_1}-1$ 和 $x^{e_2}-1$ 的公因子，据引理 2.5.1 知，$\mathrm{ord}(f)\mid e_1$ 且 $\mathrm{ord}(f)\mid e_2$，所以 $\mathrm{ord}(f)\mid (e_1,e_2)=d$，从而 $f(x)\mid x^d-1$。综上有 $f(x)=x^d-1$。

定理 2.5.20　设 $g\in F_q[x]$是 F_q 上的不可约多项式，$f=g^b$，其中 b 是一正整数，$g(0)\neq 0$。假设 $\mathrm{ord}(g)=e$，t 是满足 $p^t\geqslant b$ 的最小正整数，其中 p 是 F_q 的特征。那么 $\mathrm{ord}(f)=e\cdot p^t$。

证明：设 $c=\mathrm{ord}(f)$。由于 $f(x)\mid x^c-1\Rightarrow g(x)\mid x^c-1$，所以 $e=\mathrm{ord}(g)\mid c$。另一方面，由于 $g(x)\mid x^e-1$，所以 $f(x)=g(x)^b\mid (x^e-1)^b$。但$(x^e-1)^b\mid (x^e-1)^{p^t}=x^{ep^t}-1$，所以 $f(x)\mid x^{ep^t}-1$，所以 $c\mid ep^t$。考虑到已证明的 $e\mid c$，所以 c 具有形式 $c=ep^u$，其中 u 是一正整数，且 $0\leqslant u\leqslant t$。

根据推论 2.5.2，$(e,p)=1$，所以 x^e-1 的根都是单根，因此 $x^{ep^u}-1=(x^e-1)^{p^u}$ 的根的重数都为 p^u，但 $f(x)\mid x^{ep^u}-1$ 且 $f(x)=g(x)^b$ 的根的重数都至少是 b，所以 $p^u\geqslant b$，因此 $u\geqslant t$，从而 $u=t$，即 $c=ep^t$。

定理 2.5.21　设 $g_1,g_2,\cdots,g_k$ 是 F_q 上两两互素的非零多项式，$f(x)=g_1g_2\cdots g_k$，则 $\mathrm{ord}(f)$是 $\mathrm{ord}(g_1),\mathrm{ord}(g_2),\cdots,\mathrm{ord}(g_k)$的最小公倍数。

证明：容易看出，只要考虑 $g_i(0)\neq 0$ 的情况即可。假设 $e=\mathrm{ord}(f)$，$e_i=\mathrm{ord}(g_i)$，$i=1,2,\cdots,k$，$c=[e_1,e_2,\cdots,e_k]$。由于 $g_i(x)\mid x^{e_i}-1$，所以 $g_i(x)\mid x^c-1$，从而 $f(x)\mid x^c-1$，$e=\mathrm{ord}(f)\mid c$。另一方面，$f(x)\mid x^e-1\Rightarrow g_i(x)\mid x^e-1\Rightarrow e_i=\mathrm{ord}(g_i)\mid e$，所以 $c\mid e$。综上，证明了 $e=c$，即 $\mathrm{ord}(f)=[e_1,e_2,\cdots,e_k]$。定理得证。

实际上用上述方法还可以证明有限个多项式的最小公倍数的阶正好是这些多项式阶的最小公倍数。

例 2.5.5　计算 $f(x)=x^{10}+x^9+x^3+x^2+1\in F_2[x]$的阶。

首先 $f(x)=(x^2+x+1)^3(x^4+x+1)$。由于 $\mathrm{ord}(x^2+x+1)=3$，所以 $\mathrm{ord}((x^2+x+1)^3)=12$，另 $\mathrm{ord}(x^4+x+1)=15$，所以 $\mathrm{ord}(f)=[12,15]=60$。注意这时 $60\nmid 2^{10}-1$，从而验证了推论 2.5.2 对可约多项式并不总是成立的。

推论 2.5.3　设 F_q 是特征为 p 的有限域，$f(x)\in F_q[x]$是一正次数多项式，$f(0)\neq 0$。设$f(x)=af_1^{b_1}\cdots f_k^{b_k}$，其中 $a\in F_q$，$b_1,\cdots,b_k$ 都是正整数且 $f_1,f_2,\cdots,f_k$ 是不同的首一不可约多项式，则 $\mathrm{ord}(f)=ep^t$，其中 e 是 $\mathrm{ord}(f_1),\cdots,\mathrm{ord}(f_k)$的最小公

倍数，t 是满足 $p^t \geqslant \max(b_1, \cdots, b_k)$ 的最小正整数。

定义 2.5.8 设 $f(x) \in F_q[x]$，$\deg(f(x)) = m \geqslant 1$。如果 $f(x)$ 正好是 F_{q^m} 的某一本原元在 F_q 上的极小多项式，则称 $f(x)$ 是 F_q 上的本原多项式。

定理 2.5.22 次数为 m 的多项式 $f(x) \in F_q[x]$ 是 F_q 上的本原多项式当且仅当 $f(x)$ 是首一的，$f(0) \neq 0$ 且 $\mathrm{ord}(f) = q^m - 1$。

证明：如果 $f(x)$ 是 F_q 上的本原多项式，则 $f(x)$ 是 Fq 上的首一不可约多项式，$f(0) \neq 0$，且以某一本原元为根。因此根据定理 2.5.19，$f(x)$ 的阶等于 $f(x)$ 任一根的阶，从而等于本原元的阶，即 $\mathrm{ord}(f(x)) = q^m - 1$。

反过来，假设 $\mathrm{ord}(f(x)) = q^m - 1$，下面首先证明 $f(x)$ 是 F_q 上不可约的多项式。用反证法，假设 $f(x)$ 是可约的，则或者 $f(x)$ 可以写成不可约多项式的幂或 $f(x)$ 可以写成两个互素多项式的乘积。

在第一种情况下，有 $f(x) = (g(x))^b$，其中 $g(x) \in F_q[x]$ 是不可约多项式，$b \geqslant 2$，$g(0) \neq 0$。由定理 2.5.20 可知 $\mathrm{ord}(f(x))$ 一定可以被 F_q 的特征除尽。这与 $\mathrm{ord}(f(x)) = q^m - 1$ 矛盾。

对于第二种情况，有 $f(x) = g_1(x) g_2(x)$，$(g_1(x), g_2(x)) = 1$，其中 $g_1(x)$，$g_2(x) \in F_q[x]$ 是次数分别为 $m_1 \geqslant 1$，$m_2 \geqslant 1$ 的首一多项式。设 $e_1 = \mathrm{ord}(g_1(x))$，$e_2 = \mathrm{ord}(g_2(x))$，则 $\mathrm{ord}(f(x)) \leqslant e_1 e_2$。但有 $e_1 \leqslant q^{m_1} - 1$，$e_2 \leqslant q^{m_2} - 1$，所以 $e_1 e_2 \leqslant (q^{m_1} - 1)(q^{m_2} - 1) = q^{m_1 + m_2} - q^{m_1} - q^{m-2} + 1 < q^{m_1 + m_2} - 1 = q^m - 1$。所以 $\mathrm{ord}(f(x)) < q^m - 1$，这与假设矛盾。

综上所述，$f(x)$ 是 F_q 上的不可约多项式，从而由定理 2.5.19 知 $f(x)$ 的根的阶一定为 $\mathrm{ord}(f(x)) = q^m - 1$，所以是本原元，因此 $f(x)$ 是本原多项式。

注意上述定理中的 $f(0) \neq 0$ 只是为了排除 $q = 2$，$m = 1$ 时 $f(x) = x$ 的情况。

2.5.4 Galois 环

关于 Galois 环的理论是由 W. Krull 于 1924 年发明的。

定义 2.5.9 一个具有单位元 1 的有限环称为 Galois 环，如果其所有零因子组成的集合添加上 0 正好构成一个主理想 $(p \cdot 1)$，其中，p 是某一素数。

例 2.5.6 设 p 是一个素数，s 是一个正整数，那么剩余类环 $\mathbb{Z}_{p^s}$ 是一个 Galois 环。所有零因子添加上 0 构成的主理想为 (p)。当 $s = 1$ 时，$\mathbb{Z}_{p^s} = F_p$ 是具有 p 个元素的有限域。这时 $(p) = (0)$。

定理 2.5.23 设 R 是一个 Galois 环，其所有零因子添加上 0 构成的主理想为 $(p \cdot 1)$。则 $(p \cdot 1)$ 是 R 的唯一一个极大理想，$R/(p \cdot 1)$ 是一个含有 p^m 个元素的有限域 F_{p^m}，其中 m 是某一正整数，R 的特征是 p 的某一幂次。

证明：由于在一个有限环中，任何一个元素如果不是零因子，则一定是单位，因此 $(p \cdot 1)$ 是 R 的唯一一个极大理想。因此 $R/(p \cdot 1)$ 是域。设 ϕ 是 $R \to R/(p \cdot 1)$ 的自然同态，$\bar{r}$ 表示 $r \in R$ 在 ϕ 下的像 $\phi(r)$。则有 $p \cdot \bar{1} = \underline{p \cdot 1} = \bar{0}$，所以 $R/(p \cdot 1)$ 是一个特征为 p 的有限域。不妨设 $R/(p \cdot 1) \simeq F_{p^m}$，其中 m 是某一整数。

设 k 是环 R 的特征，则由 $k\cdot 1=0$，可得 $k\cdot\bar{1}=\underline{k\cdot 1}=\bar{0}$。因此 $p\mid k$。假设 $k=p^s l$，其中 s,l 都是正整数，且 $(p,l)=1$。如果 $l>1$，则 $a=p^s\cdot 1$ 和 $b=l\cdot 1$ 都是非零元素，且 $ab=0$。从而 $l\cdot 1\in(p\cdot 1)$，$l\cdot\bar{1}=\underline{l\cdot 1}=\bar{0}\in R/(p\cdot 1)$，所以 $p\mid l$，从而与 $\gcd(p,l)=1$ 矛盾。因此 $k=p^s$。

定理 2.5.24　设 R 是一个特征为 p^s 的 Galois 环，单位元为 1，其中 p 是一个素数。那么 R 的所有零因子添加上 0 正好构成理想 $(p\cdot 1)$，$\{r\cdot 1 : r\in\mathbb{Z}_{p^s}\}$ 是 R 的一个子环，且同构于 $\mathbb{Z}_{p^s}$。

证明：根据定理 2.5.23 可知 R 的所有零因子添加上 0 构成的理想正好为 $(p\cdot 1)$。构造映射：

$$\eta:\ \mathbb{Z}_{p^s}\rightarrow R$$
$$r\mapsto r\cdot 1$$

则容易验证 η 是一个单映射，它的像是 $\{r\cdot 1 : r\in\mathbb{Z}_{p^s}\}$。

设 R 是一个特征为 p^s 的 Galois 环。通常把 $r\in\mathbb{Z}_{p^s}$ 和元素 $r\cdot 1\in R$ 看成是一样的，而且把 $\mathbb{Z}_{p^s}$ 看成 R 的子环。特别地，$p\in\mathbb{Z}_{p^s}$ 和 $p\cdot 1\in R$ 被看成是一样的，因此理想 $(p\cdot 1)$ 也写成 (p)。

定理 2.5.25　设 R 是一个特征为 p^s 的 Galois 环。则 $R/(p)\simeq F_{p^m}$，其中 m 是某一正整数，且理想 (p^i)，$0\leqslant i\leqslant s-1$，含有 $p^{(s-i)m}$ 个元素，特别地，$|R|=p^{sm}$。

证明：根据定理 2.5.23 知 (p) 是 R 唯一的一个极大理想，因此 $R/(p)\simeq F_{p^m}$，其中 m 是某一正整数。把 R 看成一个加法群，主理想 (p^i) 看成一个子群，并考虑映射：

$$R\rightarrow (p^i)/(p^{i+1})$$
$$r\mapsto p^i r+(p^{i+1})$$

很明显这是一个群的满同态，核包含 (p)。由于核是 R 的一个理想，而且不含单位元 1，而 (p) 又是一个极大理想，因此核一定就是 (p)。根据群的同态基本定理，有群的同态：

$$R/(p)\simeq (p^i)/(p^{i+1})$$

因此

$$|R/(p)|=|(p)/(p^2)|=\cdots=|(p^{s-2})/(p^{s-1})|=|(p^{s-1})|$$

又 $R/(p)\cong F_{p^m}$，所以 $|R/(p)|=p^m$，从而

$$|(p^i)=|(p^i)/(p^{i+1})||(p^{i+1})/(p^{i+2})|\cdots|(p^{s-1})|$$

特别地，有 $|R|=|(1)|=|(p^0)|=p^{sm}$

设 R 是一个特征为 p^s 的 Galois 环。用 ϕ 表示 $R\rightarrow R/(p)$ 的自然同态映射，并用 $\bar{r}$ 表示 $r\in R$ 在 ϕ 下的像 $\phi(r)$。显然，按以下方式可把 ϕ 扩展到多项式环上：

$$R[x]\rightarrow R/(p)[x]$$
$$a_0+a_1x+\cdots+a_nx^n\mapsto\bar{a}_0+\bar{a}_1x+\cdots+\bar{a}_nx^n,\quad (a_0,\cdots,a_n\in R)$$

也把多项式 $f(x)\in R[x]$ 在上述映射下的像记为 $\bar{f}(x)$。

定理 2.5.26　设 R 是一个特征为 p^s 的 Galois 环，$|R|=p^{sm}$，其中 p 是一个素数，s 和 m 都是正整数。设 $f(x)$ 是 $\mathbb{Z}_{p^s}$ 上的一个多项式，并假设 $\bar{f}(x)$ 在域 $R/(p)\simeq$

F_{p^m} 中有一个根 $\bar{\beta}$ 满足 $\bar{f}'(\bar{\beta})\neq 0$。那么 R 中一定存在唯一的多项式 $f(x)$ 根 $\alpha\in R$，使得 $\bar{\alpha}=\bar{\beta}$。

证明：参见文献[2]。

定理 2.5.27 设 R 是一个特征为 p^s 的 Galois 环，$|R|=p^{sm}$，其中 p 是一个素数，s 和 m 都是正整数。那么 R 一定同构于环$\mathbb{Z}_{p^s}[x]/(h(x))$，其中 $h(x)$是任意一个$\mathbb{Z}_{p^s}$ 上的 m 次首一不可约多项式。

证明：参见文献[2]。

推论 2.5.4 任何两个具有相同特征和相同元素个数的 Galois 环一定是同构的。

鉴于上述推论，因此可以用符号 $GR(p^s,p^{sm})$表示任何一个具有特征 p^s 的含有 p^{sm}个元素的 Galois 环。

2.6 格

本节介绍另一类型的代数结构——格，这是 20 世纪 30 年代才引入的一个新概念，和布尔代数有着重要的联系。这里只限于介绍一些基本的知识。

2.6.1 定义和基本性质

定义 2.6.1 所谓一个半序集是一个集合 S 以及该集合的一个二元关系 $a\geqslant b$ 满足：

(1) $a\geqslant a$(对称性)；

(2) 如果 $a\geqslant b$ 且 $b\geqslant a$，则 $a=b$(反对称性)；

(3) 如果 $a\geqslant b$ 且 $b\geqslant c$，则 $a\geqslant c$(传递性)。

半序集中的一个元素 u 称为 S 的子集 A 的一个上界，如果对任何 $a\in A$，都有 $u\geqslant a$。就说一个元素 u 是 A 的一个最小上界，如果 u 是 A 的一个上界，且对 A 的任何一个上界 v，都有 $u\leqslant v$。显然，最小上界如果存在的话一定是唯一的。同样方式，也可以定义下界和最大下界。

定义 2.6.2 一个格是指这样一个半序集，其中任何两个元素都有一个最小上界和一个最大下界。

显然，一个集合对不同的偏序关系可以构成不同的格。下面就来看几个例子。

例 2.6.1 设 A 是任意非空集合，则 A 的所有子集在集合的包含关系下构成一个格，这是因为任取两个子集 S_1、S_2，则 $S_1\cap S_2$ 是 S_1 和 S_2 的最大下界，$S_1\cup S_2$ 是 S_1 和 S_2 的最小上界。

例 2.6.2 设 V 是域 F 上的向量空间。S 是 V 的所有子空间作成的集合，则 S 关于集合的包含关系作成一个偏序集。由于两个子空间和与交仍是子空间，故 S 中任意两个元都有最小上界与最大下界，从而 S 是格。这个格称为子空间格。

我们用 $a\vee b$ 表示 a 和 b 的最小上界，而用 $a\wedge b$ 表示 a 和 b 的最大下界。容易

证明 $\vee$ 和 $\wedge$ 是结合的，因此可以定义 $a_1 \vee a_2 \vee \cdots \vee a_n$ 和 $a_1 \wedge \cdots \wedge a_n$ 分别表示 $a_1, \cdots, a_n$ 的最小上界和最大下界。

定义 2.6.3 一个格称为完全格，如果其任意子集的最小上界和最大下界都存在。

在一个完全格 L 中，用 0 表示整个 L 的最小元，而用 1 表示整个 L 的最大元，则以下定理给出了一个判断一个偏序集是否是一个完全格的有用准则。

定理 2.6.1 一个具有最大元素 1 的偏序集，如果每个非空子集都有一个最大下界，则其一定是一个完全格。同样地，一个具有最小元 0 的偏序集，如果每个非空子集都有个最小上界，则其一定也是一个完全格。

证明：只证明定理的第一部分。这只需证明对任何一个非空集合 A 都有一个最小上界即可。由于 1 是 A 的一个上界，因此 A 的所有上界的集合 B 是一个非空集合，从而根据题设，存在一个 $b=\inf(B)$ 是 B 的最小元。显然，b 就是 A 的最小上界。

例 2.6.3 设 G 是一个群，则 G 的所有子群在集合的包含关系下构成一个完全格。这是因为 G 本身是一个子群，而且任意多个子群的交也是子群。

例 2.6.4 一个群 G 的所有正规子群组成的集合在集合的包含关系下构成一个完全格。实际上，一组正规子群的最小上界就是由这组群中元素生成的子群。

例 2.6.5 例 2.6.1 是一个完全格，其中 $1=A, 0=\varnothing$。

定理 2.6.2 设 L 是格，则对 L 中的任何元素 a、b 和 c，二元运算 $\vee$ 和 $\wedge$ 满足：

1)（幂等律）$a \vee a=a, a \wedge a=a$；

2)（交换律）$a \wedge b=b \wedge a, a \vee b=b \vee a$；

3)（结合律）$(a \wedge b) \wedge c=a \wedge (b \wedge c), (a \vee b) \vee c=a \vee (b \vee c)$；

4)（吸收律）$a \wedge (a \vee b)=a, a \vee (a \wedge b)=a$。

定理 2.6.3（对偶原理） 设 P 是对任意偏序集为真的一个命题，P' 是将 P 中所有的"$\geqslant$"和"$\leqslant$"交换位置得到的对偶命题，则 P' 对任意偏序集也都为真。

定理 2.6.2 的逆命题也成立。也就是说，格的概念可以不用偏序集的概念来定义，而用具有两个运算 $\vee$ 和 $\wedge$ 并满足 1)～4)这 4 条性质来定义。

格 L 的一个子集 M 称为一个子格，如果运算 $\vee$ 和 $\wedge$ 在 M 中是封闭的，显然对 L 中的任意两个固定的元素 $a \leqslant b$，集合 $S=\{x \mid a \leqslant x \leqslant b\}$ 是一个子格，就把这个子格称为区间，用 $I[a,b]$ 表示。

定义 2.6.4 设 L 和 L' 是两个格，L 到 L' 的映射 $a \rightarrow a'$ 称为一个同态，如果对所有的 $a, b \in L$，都有 $(a \vee b)'=a' \vee b'$，$(a \wedge b)'=a' \wedge b'$。一一对应的同态称为同构。

定理 2.6.4 一个从 L 到 L' 的双射是一个格同构当且仅当该双射及其逆映射都是保序的。

2.6.2 格的分配律和 Dedekind 格

定义 2.6.5 我们说格 S 是一个分配格，如果对于任意的 $a, b, c \in S$，都有

$$a \wedge (b \vee c) = (a \wedge b) \vee (a \wedge c)$$

$$a \vee (b \wedge c) = (a \vee b) \wedge (a \vee c)$$

根据对偶定理,显然上面两个分配律中只要一个成立,则另一个也成立。

例 2.6.6 所有自然数关于整除所做成的格是分配格。

定理 2.6.5 任何一个全序集 S 是一个分配格。

证明:设 $a,b,c\in S$,分成两种情况:(1) $a\geqslant b,a\geqslant c$;(2)$a\leqslant b$ 或 $a\leqslant c$。对于(1),有 $a\wedge(b\vee c)=b\vee c,(a\wedge b)\vee(a\wedge c)=b\vee c$。对于(2),有 $a\wedge(b\vee c)=a,(a\wedge b)\vee(a\wedge c)=a$。因此 S 是分配格。

定理 2.6.6 设 S 是一个分配格,$a,x,y\in S$,如果 $a\wedge x=a\wedge y$,并且 $a\vee x=a\vee y$,那么 $x=y$。

证明:$x=x\vee(x\wedge a)=x\vee(y\wedge a)=(x\vee y)\wedge(x\vee a)=(x\vee y)\wedge(y\vee a)$
$=(y\vee x)\wedge(y\vee a)=y\wedge(x\vee a)=y\wedge(y\vee a)=y$。

利用上述定理,可以很容易地判定一个格不是分配格。

定义 2.6.6 设格 S 有单位元 1 和零元 0。取 $a\in S$,若存在 $x\in S$,使得 $a\wedge x=0$,$a\vee x=1$,则称 x 是 a 的一个补元,记为 a'。

定理 2.6.7 在有单位元和零元的分配格中,一切补元的集合作成一个子格。

由于许多代数中重要的格,如一个群 G 的正规子群格,都不是分配格,因此需要将分配律的条件减弱一些,这就是将要介绍的 Dedekind 格。

定义 2.6.7 我们说格$(S,\vee,\wedge)$是一个 Dedekind 格(或模格),若对任意的 a,$b,c\in S$,只要 $b\leqslant a$,就有

$$a\wedge(b\vee c)=b\vee(a\wedge c)$$

成立。

定理 2.6.8 一个群的所有正规子群构成的格是 Dedekind 格。一个模的所有子模构成的格也是 Dedekind 格。

证明:由于任意两个正规子群 H_1 和 H_2 生成的子群为 $H_1H_2=H_2H_1$,因此只要证明对任意的正规子群 H_1、H_2、H_3,如果 $H_1\supset H_2$,则

$$H_1\cap(H_2H_3)=H_2(H_1\cap H_3)$$

对此,只需证

$$H_1\cap(H_2H_3)\subset H_2(H_1\cap H_3)$$

即可。

设 $a\in H_1\cap(H_2H_3)$,则 $a=h_1=h_2h_3,h_i\in H_i$,但 $H_1\supset H_2$,因此 $h_3=h_2{}^{-1}h_1\in H_1$。这样,$h_3\in H_1\cap H_3$,$a=h_2h_3\in h_2(H_1\cap H_3)$。从而,定理的前半部分得证。而对定理的后半部分,可以类似地证明,此处略去。

定理 2.6.9 一个格 L 是 Dedekind 格当且仅当:对任意的 $a,b,c\in L$,只要 $a\geqslant b,a\wedge c=b\wedge c,a\vee c=b\vee c$,则必有 $a=b$。

证明:必要性:设 L 是 Dedekind 格,则据定理的条件,$a=a\wedge(a\vee c)=a\wedge(b\vee c)=b\vee(a\wedge c)=b\vee(b\wedge c)=b$。

充分性:假设 L 是任何一个满足定理条件的一个格,并假设 $a,b,c\in L,a\geqslant b$。我们知道 $a\wedge(b\vee c)\geqslant b\vee(a\wedge c)$,而且

$$(a\wedge(b\vee c))\wedge c=a\wedge((b\vee c)\wedge c)=a\wedge c$$

$$a \wedge c = (a \wedge c) \wedge c \leqslant (b \vee (a \wedge c)) \wedge c \leqslant a \wedge c$$

因此

$$(b \vee (a \wedge c)) \wedge c = a \wedge c$$

上述两个关系式对任何的格都成立。因此,根据对偶定理,当 $a \geqslant b$ 时,也有

$$(b \vee (a \wedge c)) \vee c = b \vee c$$
$$(a \wedge (b \vee c)) \vee c = b \vee c$$

所以

$$(a \wedge (b \vee c)) \wedge c = (b \vee (a \wedge c)) \wedge c$$
$$(b \vee (a \wedge c)) \vee c = (a \wedge (b \vee c)) \vee c$$

从而根据定理假设,有

$$(a \wedge (b \vee c)) = (b \vee (a \wedge c))$$

定理得证。

类似于群论中的同构定理,有以下定理。

定理 2.6.10　设 L 是 Dedekind 格,$a,b \in L$,则映射 $x \to x \wedge b$ 是从区间 $I[a, a \vee b]$ 到区间 $I[a \wedge b, b]$ 上的同构映射,逆映射为 $y \to y \vee a$。

证明:首先注意到映射 $x \to x \wedge b$ 和 $y \to y \vee a$ 都是保序的。由于 $x \geqslant y \Leftrightarrow x \vee y = x \Leftrightarrow x \wedge y = y$,所以 $x \vee y = x \Rightarrow (x \vee a) \vee (y \vee a) = (x \vee y) \vee (a \vee a) = (x \vee y) \vee a = x \vee a$,因此 $x \geqslant y \Rightarrow x \vee a \geqslant y \vee a$。同样地,有 $x \wedge a \geqslant y \wedge a$。现在如果 $a \leqslant x \leqslant a \vee b$,那么 $a \wedge b \leqslant x \wedge b \leqslant b = (a \vee b \wedge b$,而且如果 $a \wedge b \leqslant y \leqslant b$,那么 $a = a \vee (a \wedge b) \leqslant y \vee a \leqslant a \vee b$。因此映射 $x \to x \wedge b$ 和映射 $y \to y \vee a$ 分别把 $I[a, a \vee b]$ 映射到 $I[a \wedge b, b]$,把 $I[a \wedge b, b]$ 映射到 $I[a, a \vee b]$。由于这两个映射都是保序的,因此只要能证明这两个映射是互逆的即可。

设 $a \in I[a, a \vee b]$,则 $x \geqslant a$,根据 Dedekind 格的定义,

$$(a \wedge b) \vee a = x \wedge (a \vee b)$$

而 $x \leqslant a \vee b$,因此 $(x \wedge b) \vee a = x$。对偶地,可以证明,如果 $y \in I[a \wedge b, b]$,也有 $(y \vee a) \wedge b = y$。

下面介绍关于 Dedekind 格的一个重要定理,它将群论以及环论中重要定理用统一的方法来处理。通过这个定理,可以看到格在代数中的作用。

定义 2.6.8　设 L 是一个格,$a,b \in L$,$a < b$。如果 L 中不存在 x,使得 $a < x < b$,则说 b 覆盖 a。

定义 2.6.9　设 L 是一个有零元的 Dedekind 格,$x \in L$。若 L 存在有限序列 $x = x_0, x_1, \cdots, x_d = 0$,使得

$$x = x_0 > x_1 > \cdots > x_d = 0$$

并且 x_i 覆盖 x_{i+1},$i = 0, 1, \cdots, d-1$,则说上式是 x 到 0 的一个极大链,d 是它的长度。x 到 0 所有极大链的最大长度叫做 x 的维数,记做 $d(x)$。

定理 2.6.11　L 是一个有零元的 Dedekind 格,$x \in L$。若 x 的维数 $d(x)$ 有限,则 x 到 0 的每一极大链的长度都相等。

证明:对 $d(x)$ 用数学归纳法证明。当 $d(x) = 1$ 时,则 x 到 0 的链仅有 $x > 0$,定

理成立。现假设 $d(x)=d-1$ 时定理成立，我们来看 $d(x)=d$ 的情形。设 x 到 0 有两个极大链：

$$x = x_0 > x_1 > \cdots > x_d = 0$$
$$x = y_0 > y_1 > \cdots > y_s = 0$$

如果 $x_1=y_1$，则由归纳假定，$s-1=d-1$，因此 $s=d$。如果 $x_1\neq y_1$，命 $z_2=x_1\wedge y_1$，则 $I[x,x_1]\cong I[y_1,z_2]$，又因 x 覆盖 x_1，所以 y_1 覆盖 z_2。又 $I[x,y_1]\cong I[x_1,z_2]$，x 覆盖 y_1，所以 x_1 覆盖 z_2。由于 $d(x)=d$，故 $d(x_1)=d-1$，根据归纳假设，x_1 到 0 的所有极大链的长度都相同，均为 $d-1$，从而 z_2 到 0 的极大链的长度均为 $d-2$。再由 $d(x)=d$，以及 y_1 经由 z_2 到 0 的长度为 $d-1$，知 $d(y_1)=d-1$。根据归纳假设，y_1 到 0 的任意极大链均有相同的长度 $d-1$，于是

$$y_1 > y_2 > \cdots > y_s = 0$$

的长度也为 $d-1$。所以 $s=d$。

定理 2.6.12（维数定理） L 是一个有零元的 Dedekind 格，$x,y\in L$。若 $d(x)$、$d(y)$均有限，则

$$d(x) + d(y) = d(x \vee y) + d(x \wedge y)$$

证明：在 L 中，由于 $I[x\vee y,x]$和 $I[y,x\wedge y]$同构，因此维数相同，而 $d(I[x\vee y,x])=d(x\vee y)-d(x)$，$d(I[y,x\wedge y])=d(y)-d(x\wedge y)$，所以

$$d(x\vee y)-d(x)=d(y)-d(x\wedge y)$$

从而

$$d(x)+d(y)=d(x\vee y)+d(x\wedge y)$$

将上述维数定理用于由向量空间的所有子空间构成的格上，就可以得到熟悉的关于向量空间的维数定理。维数定理是 Dedekind 格的一个重要特征，也就是说上述定理的逆命题也成立。

定理 2.6.13 设 L 是一个有零元的格，且每一元的维数都有限。如果对任意的 $x,y\in L$，有

$$d(x)+d(y)=d(x\vee y)+d(x\wedge y)$$

那么 L 是一个 Dedekind 格。

证明：设 $a,b,c\in L$，$a\geqslant b$，希望证明

$$a\wedge(b\vee c)=b\vee(a\wedge c)$$

因为 $b\leqslant a$，$b\leqslant b\vee c$，所以 $b\leqslant a\wedge(b\vee c)$。又 $a\wedge c\leqslant a$，$a\wedge c\leqslant c\leqslant b\vee c$，从而 $a\wedge c\leqslant a\wedge(b\vee c)$，$b\vee(a\wedge c)\leqslant a\wedge(b\vee c)$。利用维数关系，有

$$\begin{aligned}
d(b\vee(a\wedge c)) &= d(b)+d(a\wedge c)-d(b\wedge(a\wedge c))\\
&= d(b)+d(a\wedge c)-d(b\wedge c)\\
&= d(b\vee c)-d(c)+d(a\wedge c)\\
&= d(b\vee c)+d(a)-d(a\vee c)\\
&= d(a)+d(b\vee c)-d(a\vee(b\vee c))\\
&= d(a\wedge(b\vee c))
\end{aligned}$$

从而 $b\vee(a\wedge c)=a\wedge(b\vee c)$，因此 L 是 Dedekind 格。

2.7　基本方法与应用举例

2.7.1　快速指数运算

设 G 是一个具有乘法运算的群，给定 G 中的一个元素 g（通常 g 是固定的）和一个正整数 a（通常 a 是变化的），求 g^a 称为 G 中的指数运算。在密码应用中，G 通常是一个有限群，因此容易证明存在正整数 n 使得 $g^n=1$。这样，取 a 模 n 的余数 $b<n$，则有 $g^a=g^b$。因此，总可以将指数运算限定为指数小于 n 的情形。

计算 g^n 的一个平凡方法是依次计算

$$g^2 = g \cdot g, g^3 = g^2 \cdot g, g^4 = g^3 \cdot g, \cdots, g^a = g^{a-1} \cdot g$$

这种计算方法的复杂度显然是 $O(n)$。当 n 很大时（如在密码应用中，$n \leqslant 2^{60}$），这种方法不可行。

重复平方－乘方法（又称二元方法）是将复杂度从 $O(n)$ 降为 $O(\log_2 n)$ 的典型方法。设 $r=\lceil \log_2 n \rceil$ 为不小于 $\log_2 n$ 的正整数，即 r 是 n 的二进制长度。不妨假定 $0 \leqslant a < n$。将 a 二进制展开得到

$$a = a_r \cdot 2^r + a_{r-1} \cdot 2^{r-1} + \cdots + a_1 \cdot 2 + a_0, \quad a_i \in \{0,1\}$$

依次平方可得 $g^2, g^{2^2}, g^{2^3}, \cdots, g^{2^r}$，再将那些对应到 $a_i=1$ 的 g^{2^i} 依次相乘，即得 $\Pi_{a_i=1} g^{2^i} = g^{\sum a_i 2^i} = g^a$。这种方法的计算复杂度是 $O(r)=O(\log_2 n)$ 次基本运算（平方或乘法运算）。

上述运算是从 a 的二进制表示的低位逐次往高位计算，也可以从高位逐次往低位计算：由于 a 可以写成

$$a = 2 \cdot (\cdots \cdot 2 \cdot (2 \cdot (2 \cdot a_r + a_{r-1} + a_{r-2}) + \cdots + a_1) + a_0$$

因此

$$g^a = (\cdots \cdot (((g^{a_r})^2 \cdot g^{a_{r-1}})^2 \cdot g^{a_{r-2}})^2 \cdot \cdots \cdot g^{a_1})^2 \cdot g^{a_0}$$

显然，从高位到低位的计算方法与从低位到高位的计算方法具有相同的复杂度，它们具有同样多的乘法基本运算（对那些 $a_i=1$ 的 i）和同样多的平方运算（r 次）。但后者的平方运算是针对固定元素的，在可预计算并存储 g^{2^i} $(1 \leqslant i \leqslant r)$ 的计算环境里，这些平方运算的计算量可以不计。

重复平方-乘方法有许多变型，包括窗口法、滑动窗口法、带负号二进制法和 Shamir 方法等。

1. 窗口法

将 a 的二进制表示 $(a_r a_{r-1} \cdots a_1 a_0)$ 按长度为 w 分成 $k=\left\lceil \frac{r+1}{w} \right\rceil$ 块。每块为 w 个比特，表示从 0 到 2^w-1 中的一个数。设这 k 个数依次为 $b_{k-1}, \cdots, b_1, b_0$，则

$$a = b_{k-1} \cdot 2^{(k-1)w} + \cdots + b_1 \cdot 2^w + b_0$$

$$g^a = \prod_i (g^{2^{wi}})^{b_i}$$

其中，$g^{2^{wi}}$是 g 连续作 i 个 2^w 次升幂得到的，g^{b_i} 可以通过查阅预先计算的表$\{g,g^2,\cdots,g^{2^w-1}\}$得到。若存储受限，这个预计算表还可以只存储奇指数次幂，即$\{g,g^3,\cdots,g^{2^w-1}\}$，偶指数幂可通过相应的奇数幂的平方得到。通常 w 取比较小的整数，如 $w=2,3,4,5$。

2. 滑动窗口法

在窗口法中，若有一些分块的起始比特为 0，可以将此分块往后滑动，重新构建分块。例如，原分块为(取 $w=4$)

$$1101001001000111$$

滑动窗口法分块为

$$\underline{110100}\ \underline{1001000}\ \underline{111}$$

滑动窗口法有可能减少分块的个数。

3. 带负号二进制法

对于某些群，如椭圆曲线的点群(详见定义 3.1.1)，群元素的逆可能非常容易求得，这时可以将 a 的二进制表示改变为

$$a=\sum a_i 2^i,\quad a_i\in\{-1,0,1\}$$

则

$$g^a=\Big(\prod_{a_i=1} g^{2^i}\Big)\cdot\Big(\prod_{a_i=-1}(g^{2^i})^{-1}\Big)$$

这种方法称为带负号二进制法。当 $i\geqslant 2$ 时，利用 $2^{i-1}+2^{i-2}+\cdots+2+1=2^i-1$，有

$$(11\cdots1)_2=(10\cdots0-1)_2 \tag{2.1}$$

利用这个关系，可以根据 a 的二进制表示得到其他带负号二进制表示。例如：

$$a=(11010111110001011101)_2$$

从低位开始，按照式(2.1)的关系逐步将连续的比特 1 进行替换，得到其带负号的二进制表示

$$a=(10-101-1000-10010-100-101)_2$$

容易证明，带负号的二进制表示的长度最多比二进制长度大 1。同时，由于$(10-1)_2=(11)_2$，$(1-1)_2=(01)_2$，带负号的二进制有时还可以化成零系数比较多的情况。如上述例子，

$$a=(11001000-10001100-101)_2$$

数学上已经证明，长度为 r 的整数的二进制表示中平均有$\frac{1}{2}$的系数为 0，而带负号的二进制表示中平均有$\frac{2}{3}$的系数为 0。这有助于减少指数运算的乘法操作次数。

4. Shamir 方法

在某些密码算法和安全协议中(如 DSA 标准)，需要计算双指数运算 $g^a h^b$，其中 g,h 是群中两个固定元素，a、b 是两个变化的整数。

可以先分别计算 g^a 和 h^b，然后将它们相乘得到 $g^a h^b$ 的结果。但这样计算的复

杂度较大。Shamir 方法则直接将两个指数运算合并成一个运算，从而提高计算速度。

例 2.7.1　计算 $g^{37}h^{20}$ 如下：

37	=	(1	0	0	1	0	$1)_2$
20	=	(0	1	0	1	0	$0)_2$
^{2}		1	g^2	g^4h^2	g^8h^4	$g^{18}h^{10}$	$g^{36}h^{20}$
$\cdot g$		g					$g^{37}h^{20}$
$\cdot h$			g^2h				
$\cdot gh$					g^9h^5		

从以上可以看出，Shamir 方法的实质是将两个指数运算 g^a 和 h^b 中的分别平方运算合并成一个平方运算，同时将乘以 g 再乘以 h 的运算合并成一次乘以 gh 的运算（预先算好 gh）。对于 r 比特的整数 a 和 b，这样可以平均节省 r 个平方运算和 $\frac{r}{4}$ 个乘法运算。

Shamir 方法也有窗口法、带负号系数等变型版本。

2.7.2　Gröbner 基

Gröbner 基是奥地利数学家 Buchberg 于 20 世纪 60 年代发明的，是符号计算、计算机代数等领域中的重要理论与方法，在多元多项式方程组求解、理想成员判定等方面有着广泛的应用。本节将简要介绍 Gröbner 基理论与相关算法。

设 F 是域，$F[X_1,X_2,\cdots,X_n]$是域 F 上未定元 $X_1,X_2,\cdots,X_n$ 的多变元多项式环。未定元 $X_1,X_2,\cdots,X_n$ 的一个项 t 是指以下形式的幂积 $X_1^{e_1}X_2^{e_2}\cdots X_n^{e_n}$，其中 $e_i\in\mathbb{N}$，$1\leqslant i\leqslant n$。特别地，定义 $1=X_1^0\cdots X_n^0$。用 $T(X_1,X_2,\cdots,X_n)$或 T 表示所有项组成的集合。在 T 中，可以定义项序如下。

定义 2.7.1　T 上的一个项序是这样一种全序$\leqslant$，满足：

1）对所有 $t\in T$，有 $1\leqslant t$ 成立；

2）如果 $t_1,t_2\in T$，且 $t_1\leqslant t_2$，那么对所有的 $s\in T$，有 $st_1\leqslant st_2$。

容易验证，对于 T 上的项序，有以下定理。

定理 2.7.1　设$\leqslant$是 T 上的一个项序，则

1）任给 $s,t\in T$，如果 $s|t$，则必有 $s\leqslant t$ 成立。

2）设 $s_1,t_1,s_2,t_2\in T$，$s_1\leqslant s_2$，$t_1\leqslant t_2$，则 $s_1t_1\leqslant s_2t_2$。

例 2.7.2　以下每个都是 T 的项序。

1）字典序：$x_1^{d_1}\cdots x_n^{d_n}<x_1^{e_1}\cdots x_n^{e_n}\Leftrightarrow\exists i$ 使得 $d_i<e_i$，但 $1\leqslant j<i$ 时，$d_j=e_j$。

2）反字典序：$x_1^{d_1}\cdots x_n^{d_n}<x_1^{e_1}\cdots x_n^{e_n}\Leftrightarrow\exists i$ 使得 $d_i<e_i$ 但 $i<j\leqslant n$ 时，$d_j=e_j$。

3）全次数字典序：$x_1^{d_1}\cdots x_n^{d_n}<x_1^{e_1}\cdots x_n^{e_n}\Leftrightarrow\sum_{i=1}^{n}d_i<\sum_{i=1}^{n}e_i$ 或者 $\sum_{i=1}^{n}d_i=\sum_{i=1}^{n}e_i$ 但在字典序下 $x_1^{d_1}\cdots x_n^{d_n}<x_1^{e_1}\cdots x_n^{e_n}$。

4）全次数反字典序：$x_1^{d_1}\cdots x_n^{d_n}<x_1^{e_1}\cdots x_n^{e_n}\Leftrightarrow\sum_{i=1}^{n}d_i<\sum_{i=1}^{n}e_i$ 或者 $\sum_{i=1}^{n}d_i=\sum_{i=1}^{n}e_i$ 但

在反字典序下 $x_1^{d_1}\cdots x_n^{d_n}<x_1^{e_1}\cdots x_n^{e_n}$。

5）分块序：把变元分成两组或多组，先比较第一组的序，如果相等再比较第二组的序，依次类推。

定理 2.7.2 设$\leqslant$是一项序，那么任意一多项式 $f\in F[X_1,X_2,\cdots,X_n]$可以唯一地表示为

$$f=\sum_{i=1}^{k}a_i m_i$$

其中 $a_i\in F, m_i\in T$，且 $m_1>\cdots>m_k$。就称上述表达式中的 m_1 为多项式 f 的首项，记为 $HT(f)$，a_i 为 f 的首项系数，记为 $HC(f)$，而单项式 $HM(f)=HC(f)\cdot HT(f)$称为 f 的头单项式。

定义 2.7.2（多项式的约化） 设$\leqslant$是一项序，$f,g\in F[X_1,X_2,\cdots,X_n]$且 $g\neq 0$。如果存在 $t\in T(f)$，$s\in T$ 使得 $t=s\cdot HT(g)$，则称 f 是模 g 可约化的。这时，令

$$p=f-\frac{c}{HC(g)}\cdot s\cdot g$$

其中 c 是 f 关于 t 的系数，并说 f 通过模 g 消去 t 一步约化到 p，记做 $f\xrightarrow[g]{t}p$，或简记为 $f\xrightarrow[g]{}p$。

设 $G\subset F[X-1,\cdots,X_n]$是一多项式组。如果存在一多项式 $g\in G$ 使得 $f\xrightarrow[g]{}p$，则称 f 是模G 可约化的，记为 $f\xrightarrow[G]{}p$。如果没有这样的 g 存在，则称 f 对G 为已约化的，或为范式。显然，若 f 对G 不是已约化的，则一定可以在有限步约化之后，得到以下的余式公式：

$$f=\sum_{i=1}^{s}q_u g_i+r$$

其中 $g_i\in G$，$r\in F[X_1,X_2,\cdots,X_n]$，且 r 对G 是已约化的，则称 r 为 f 模G 的余式或范式，记为 $f\xrightarrow[G]{*}r$。由 f 和G 求得 r 的过程称为 f 对G 的约化。

定义 2.7.3（Gröbner 基） 设 $G\subset K[X_1,X_2,\cdots,X_n]$是一多项式的有限集，$0\notin G$。如果在项序$\leqslant$下，对由 G 生成的理想 $Id(G)$中的任一多项式 f，都存在 $g\in G$ 使得 $HT(g)\mid HT(f)$，则称 G（相对于项序$\leqslant$）是 Gröbner 基，也称 G 是理想 $Id(G)$的 Gröbner 基。

定理 2.7.3 设 $I\subset K[X_1,X_2,\cdots,X_n]$是一理想，$G$ 是 I 的一非空子集，$0\notin G$。则下列条件等价：

1）G 是 I 的 Gröbner 基；

2）I 中任意非零多项式都是模G 可约化的；

3）对任意的 $f\in I$，有 $f\xrightarrow[G]{*}0$；

4）I 中任意非零多项式都可写成 $f=\sum_{i=1}^{t}h_i g_i$ 的形式，其中，$h_1,h_2,\cdots,h_t\in$

$F[X_1,X_2,\cdots,X_n]$，$HT(f)=\max\{HT(h_i)HT(g_i)\mid i=1,2,\cdots,t\}$；

5）$\langle HT(G)\rangle=\langle HT(I)\rangle$。

证明：1）⇒2）：任给 I 中的非零多项式 f，根据定义可知存在 $g\in G$，使得 $HT(g)\mid HT(f)$，因此 f 是模 G 可约化的。

2）⇒3）：设 $f\in I$，f 模 G 的范式为 r，则根据范式的定义可知 r 是模 G 已约化的多项式。但根据约化的定义又知，$r\in I$，从而根据题设，r 只能是 0，否则 r 是模 G 可约化的，与其是模 G 的范式相矛盾。

3）⇒4）：显然.

4）⇒5）：明显有 $\langle\{HT(g)\mid g\in G\}\rangle\subset\langle\{HT(f)\mid f\in I\}\rangle$。另一方面，对任意的 $f\in I$，由 3)可知 $f\in I$ 当且仅当 $f=\sum_{i=1}^{t}h_ig_i$ 和 $HT(f)=\max\{HT(h_i)HT(g_i)\mid i=1,2,\cdots,t\}$，于是 $HT(f)=\sum_{HT(f)=HT(h_i)HT(g_i)}HT(h_i)HT(g_i)$，这表明 $HT(f)\in\langle\{HT(g)\mid g\in G\}\rangle$。

5）⇒1）：由题意，对任意的 $f\in I$，有 $HT(f)=\sum_{HT(f)=HT(h_i)HT(g_i)}HT(h_i)HT(g_i)$ 对某些 $h_i\in K[x_1,x_2,\cdots,x_n]$和 $g_i\in G$ 成立，因此必有 $HT(g_i)\mid HT(f)$对某一 $g_i\in G$ 成立，所以 G 是 I 的 Gröbner 基。

定义 2.7.4 设 $f,g\in F[X_1,X_2,\cdots,X_n]$是两个非零多项式，定义 f 和 g 的 S 多项式为

$$\operatorname{spol}(f,g)=HC(g)\cdot\frac{t}{HT(f)}\cdot f-HC(f)\cdot\frac{t}{HT(g)}\cdot g$$

其中 $t=\operatorname{lcm}(HT(f),HT(g))$。

定理 2.7.4 设 G 是一多项式的非空集合，$0\notin G$，则 G 是一 Gröbner 基当且仅当对任意的 $f,g\in G$，都有 $\operatorname{spol}(f,g)\xrightarrow[G]{*}0$。

算法 2.7.1（Buchberger 算法） 输入：一个有限多项式集合 $F\subset F[X_1,X_2,\cdots,X_n]$。

输出：理想 $Id(F)$的 Gröbner 基。

```
begin
G←F
B←{{g1,g2}|g1,g2∈G,g1≠g2}
while B≠∅ do
从 B 中选取{g1,g2}
B←B\{{g1,g2}}
h←spol(g1,g2)
h0←h 模 G的一个范式
if h0≠0 then
    B←B∪{{g,h0}|g∈G}
    G←G∪{h0}
  end
```

```
end
end
```

2.7.3 Ritt-吴特征列方法

Ritt-吴特征列方法是由我国著名数学家吴文俊先生于 20 世纪 70 年代发明的，是多变元多项式方程组求解、多变元多项式组零点结构分析的重要工具与算法。本节将简要介绍 Ritt-吴特征列方法基本理论与算法。

设 K 为一域，$K[x_1,x_2,\cdots,x_N]$是 K 上以 $x_1,x_2,\cdots,x_N$ 为变元的所有多项式组成的多项式环，$K(x_1,x_2,\cdots,x_N)$是 $K[x_1,x_2,\cdots,x_N]$的分式域。对任意多项式 $f\in K[x_1,x_2,\cdots,x_N]$，用 $\deg_{x_i}f$ 表示变元 x_i 在多项式 f 中实际出现的最高次数。不出现任何变元的多项式称为常数多项式。

假设变元 $x_1,x_2,\cdots,x_N$ 有次序关系 $x_1<x_2<\cdots<x_N$。对 $K[x_1,x_2,\cdots,x_N]$中任一多项式 f，f 的类 $\mathrm{cls}(f)$是一非负整数，定义为 f 中实际出现的变元的最大下标。如果 f 为常数多项式，则定义为 0。

假设 $f\in K[x_1,x_2,\cdots,x_N]$，$\mathrm{cls}(f)=m\neq 0$，则 f 可以写为

$$f = x_m^d + x_m \text{ 的低次项}$$

其中 $d=\deg_{x_m}f$，$I\in k[x_1,x_2,\cdots,x_{m-1}]$。一般来说，$I$ 是 $x_1,x_2,\cdots,x_{m-1}$ 的多项式，则称 I 为多项式 f 的初式，记做 $\mathrm{Init}(f)$。x_m 有时也被称为多项式 f 的主变元。在以后的讨论中，多项式的初式有着特殊的意义。

设 $f\in k[x_1,x_2,\cdots,x_N]$，$\mathrm{cls}(f)=m>0$，$K[x_1,x_2,\cdots,x_N]$中任一多项式 g 称为对 f 已约化的多项式，如果 $\deg_{x_m}g<\deg_{x_m}f$。显然，多项式 f 的初式对 f 已约化。

定理 2.7.5（伪除法） 设 $f\in K[x_1,x_2,\cdots,x_N]$为一非常数多项式，初式为 I，则对任一多项式 $g\in K[x_1,x_2,\cdots,x_N]$，总存在非负整数 s 及多项式 $Q,R\in K[x_1,x_2,\cdots,x_N]$，使得

$$I^s g = Qf + R$$

且 R 对 f 已约化。

当 s 为能有上述形式的最小非负整数时，则称 R 为 g 对 f 的余式，记做 $\mathrm{prem}(g,f)$。从 g 得到 R 的过程称为 g 对 f 的约化。显然，任一非零多项式对其自身的余式为 0。约定任一多项式对一非零常数多项式的余式总为 0。

定义 2.7.5 设 $\mathcal{AC}$ #：$f_1,f_2,\cdots,f_r$ 为一多项式的有限序列。我们称 $\mathcal{A}$为一升列如果

1）$r=1$ 但 $f_1\neq 0$；或

2）$r>0$，$0<\mathrm{cls}(f_1)<\mathrm{cls}(f_2)<\cdots<\mathrm{cls}(f_r)$且对任意的 $j>i$，f_j 对 f_i 已约化。

显然，升列的长度 r 总不大于 N。有时把单一非 0 常数多项式组成的升列称为矛盾列。

设 $f_1,f_2,\cdots,f_r$ 是一升列，任一多项式 g 对该升列的余式 $\mathrm{prem}(g,f_1,f_2,\cdots,f_r)$归纳地定义为 $\mathrm{prem}(\mathrm{prem}(g,f_r),f_1,f_2,\cdots,f_{r-1})$，令其为 R，则有公式：

$$I_1^{s_1}\cdots I_r^{s_r}g = Q_1f_1+\cdots+Q_rf_r+R$$

其中，I_i 是 f_i 的初式；s_i 为非负整数；$Q_1,\cdots,Q_r$ 都是 $K[x_1,x_2,\cdots,x_N]$中的多项式且 R 对诸 f_i 都已约化。

上述公式称做余式公式。从 g 和升列 $f_1,f_2,\cdots,f_r$ 出发得到余式 R 的过程叫做 g 对升列 $f_1,\cdots,f_r$ 的约化。

有了多项式升列的概念，就可以进一步引入多项式组的特征列。首先看下面的定义。

定义 2.7.6　设 f 和 g 是 $K[x_1,\cdots,x_n]$中的两个非 0 多项式，则称 f 比 g 有较低的秩或 g 比 f 有较高的秩，记做 $f<g$ 或 $g>f$。如果：

1）$\text{cls}(f)<\text{cls}(g)$ 或

2）$\text{cls}(f)=\text{cls}(g)=p\neq 0$，且 $\deg_{x_p}f<\deg_{x_p}g$。

当两多项式 f 和 g 不能比较秩的高低时，则称 f 和 g 有相同的秩，记做 $f\sim g$。

显然，对任一多项式的非空集合，按秩的高低，总存在一极小元素，即秩不高于其他任何多项式的多项式。

定义 2.7.7　设 $\mathcal{A}$C＃：$f_1,\cdots,f_r$；$\mathcal{B}$：$g_1,\cdots,g_s$ 为两多项式升列，则称 $\mathcal{A}$ 比 $\mathcal{B}$ 有较高的秩或 $\mathcal{B}$ 比 $\mathcal{A}$ 有较低的秩，记做 $\mathcal{A}>\mathcal{B}$ 或 $\mathcal{B}<\mathcal{A}$，如果：

1）存在 $j\leqslant\min(r,s)$，使得 $g_1\sim f_1,\cdots,f_{j-1}\sim g_{j-1},f_j>g_j$ 或

2）$s>r$ 且 $f_1\sim g_1,\cdots,f_r\sim g_r$。

在两个升列 $\mathcal{A}$ 和 $\mathcal{B}$ 不能比较秩的高低时，则称其有相同的秩，记做 $\mathcal{A}\sim\mathcal{B}$。

显然，升列对秩的高低来说构成一部分次序，因此，对一个升列的集合，可以引入极小升列的概念，即集合中秩不高于其他任何升列的升列。

定理 2.7.6　设 $\mathcal{A}_1,\mathcal{A}_2,\cdots,\mathcal{A}_q,\cdots$是一升列的非增秩序列，则一定存在 q 使得 $\mathcal{A}_q,\mathcal{A}_{q+1},\cdots$都具有相同的秩。

从上述定理出发，则有以下推论。

推论 2.7.1　对任一升列的非空集合，一定有极小升列存在。

一个升列 $\mathcal{A}$ 称为属于多项式集合 PS 的，如果 $\mathcal{A}$ 中每一多项式都是 PS 中的多项式。由于任一非零多项式都单独构成一升列，因此属于非空多项式集合 PS 的升列一定存在，故属于 PS 的升列构成一非空集合。由推论 2.7.1，从而一定有极小升列。任一这样的极小升列，都称为 PS 的一个基列。

综上所述，则有以下推论。

推论 2.7.2　设 PS 为一多项式的非空集合，则 PS 必有基列存在。

对于多项式集合的基列，则有以下定理。

定理 2.7.7　设 PS 为一多项式的非空集合，$\mathcal{A}$ 为其一基列。假设 g 是一对 $\mathcal{A}$ 已约化的非 0 多项式，则 $PS\cup\{g\}$的基列比 $\mathcal{A}$ 必有较低的秩。

设 PS 为一多项式的非空集合，结合定理 2.7.6 和定理 2.7.7 考察以下步骤：

$PS_1=PS$	$PS_2=PS_1\cup RS_1$		$PS_n=PS_{n-1}\cup RS_{n-1}$	
BS_1	BS_2	$\cdots$	BS_n	$\cdots$
RS_1	RS_2		RS_n	

其中 BS_i 为 PS_i 的基列，RS_i 是 PS_i 中多项式对 BS_i 的所有非 0 余式组成的集合。

根据定理 2.7.7,可知 $BS_1, BS_2, \cdots$ 是一升列的非增秩序列,故一定存在一正整数 m 使得 $BS_m \sim BS_{m+1} \sim BS_{m+2} \sim \cdots$,这时将有 $RS_m = RS_{m+1} = \cdots = \varnothing$,也就是说 PS 中的任一多项式都将被 BS_m 约化成 0。

综上所述,则有以下定理。

定理 2.7.8 设 PS 为一多项式的非空集合,有一机械化算法可在有限步内求得一升列 CS: $f_1, f_2, \cdots, f_r$ 使得 $f_i \in \text{Ideal}(PS)$ 且对任一多项式 $f \in PS$, $\text{prem}(f, CS) = 0$。

上述定理中的 CS 叫做多项式集合 PS 的特征列。从 PS 出发求特征列 CS 的步骤叫做多项式组 PS 的整序。整序是吴方法中的一个重要算法,它把一堆杂乱无章的多项式整理成了一组具有良好性质的多项式,为进一步研究多项式组的性质及其零点结构提供了强有力的条件。

设 E 是 K 的一个扩域。用 $\text{Zero}(S)$表示 S 中的多项式在 E 上的公共零点组成的集合,即

$$\text{Zero}(S) = \{(a_1, \cdots, a_N) \in E^N \mid h(a_1, \cdots, a_N) = 0 \ \forall h \in S\}$$

假设 G 是另一多项式集合, $\text{Zero}(S/G) = \text{Zero}(S)\text{Zero}(G)$,表示是 S 的但不是 G 的所有零点所组成的集合。对于单一多项式 Q, $\text{Zero}(S/\{Q\})$有时也写为 $\text{Zero}(S/Q)$。

定理 2.7.9(整序原理) 设 PS 为一多项式的非空集合, CS 是 PS 的一特征列,则

$$\text{Zero}(CS/J) \subset \text{Zero}(PS) \subset \text{Zero}(CS)$$

$$\text{Zero}(PS)C\# = \text{Zero}(CS/J) \cup_i \text{Zero}(PS \cup \{I_i\})$$

其中, I_i 是 CS 中多项式的初式; J 为所有 I_i 的乘积。

显然,当所求得的特征列 CS 为一矛盾列时,就有 $\text{Zero}(PS) = \text{Zero}(CS) = \varnothing$。这说明 PS 中的多项式没有公共的零点。在上述定理中,如果对诸 $\text{Zero}(PS \cup \{I_i\})$ 继续进行分解,可得以下定理。

定理 2.7.10(零点分解定理) 对任一非空多项式集合 PS 有分解:

$$\text{Zero}(PS) = \bigcup_i \text{Zero}(\text{ASC}_k / J_k)$$

其中, ASC_k 是升列; J_k 为 ASC_k 中多项式的初式的乘积。更一般地,对任一多项式 G,有:

$$\text{Zero}(PS/G) = \bigcup_i \text{Zero}(\text{ASC}_k / J_k G_k)$$

其中 G_k 是 G 对 ASC_k 的余式且 $G_k \neq 0$。

在上述定理中,对某一 ASC_k,若 $G_k = 0$,显然有 $\text{Zero}(\text{ASC}_k / J_k G_k) = \varnothing$ 是一空分支,因此可以去掉。

2.7.4 有限域上的离散对数

离散对数问题被认为是一个“难”问题,而被广泛用于设计密码学原子构件(Cryptographic primitive)。例如,整数的模指数运算是密码学中常用的一种单向函数:模指数运算就是要计算表达式 $a^x \bmod n$ 的值,而这是很容易的。但模指数运算的逆问题,即给定正整数 a 和 b,求解 x 使得 $a^x \equiv b \bmod n$,却是求解一个整数的离散

对数，是一个困难问题。目前密码设计者对两种离散对数问题很感兴趣，一种是有限域乘法群中的离散对数问题，另一种是有限域上椭圆曲线上的离散对数问题。正是由于许多密码算法和协议的安全性基于离散对数问题，因此对离散对数问题的研究受到了广泛重视。在这一章将介绍有限域上的离散对数问题以及一些著名的求解离散对数的计算算法。当然，应该知道的是，目前还没有求解一般离散对数问题的有效方法。

首先给出有限域上离散对数问题的定义。我们知道，有限域的乘法群一定是循环群，所谓有限域上的离散对数问题实际上就是有限域乘法群中的离散对数问题，具体来说有以下定义。

定义 2.7.8　设 F_q 是一具有 q 个元素的有限域，g 是 F_q 的一个本原元。有限域 F_q 中的离散对数问题是指：给定 F_q 中的一个非零元素 h，求解正整数 t，使得

$$h = g^t$$

则把 t 叫做 h（相对于本原元 g）的离散对数，记做 $t=\log_g(h)$。有时人们也把 h 的离散对数称为 h 的指数。

由于阶相同的循环群一定是同构的，很难理解为什么离散对数问题在有些循环群中是困难的，而在另外一些循环群中却是容易的。一般认为，离散对数的困难性来自于群的表示方法，循环群表示方法的不同才导致了其中的离散对数问题的难易程度不同，也就是说某些循环群的表示方法隐藏了循环群的结构，从而提高了其中的离散对数问题求解的困难性。有限域的乘法群被认为很好地隐藏了循环群的结构。有限域中的离散对数问题作为一种难处理的问题，被广泛应用于密钥体制和安全协议的设计中。

例如，在许多计算机系统中，用户口令都被存在一个特殊的文件中，一旦一个人获得了阅读这个特殊文件的权限，他就可以冒充任何一个合法用户使用计算机。因此，这个特殊文件受到系统的特别保护。但事实上，如果并不直接存储用户口令本身，可以避免对这个特殊文件进行保密。一种不直接存储用户口令的方法就是使用单向函数，即这样一种函数 f，对给定的 x，很容易计算 $y=f(x)$，但对给定的 y，计算 x 使得 $f(x)=y$ 却是困难的。取代直接存储用户口令的文件，我们产生一个文件包含数组 $(i,f(p_i))$，其中 i 表示用户登录时的用户名，p_i 表示该用户的口令。这个文件可以公开。该方案的安全性依赖于单向函数 f 求逆的困难性。早期，这种单向函数选用的就是离散求幂运算，即在一个有限域 F_q 中取一个本原元 g 并公开，对任意一个整数 x，定义

$$f(x) = g^x$$

任何一个想冒充用户 i 登录计算机系统的人必须从 g^{p_i} 找出 p_i，而这相当于求解有限域 F_q 中的离散对数问题。

又例如，在密码学中，密码算法的安全性通常被设计成仅仅依赖于密钥的保密性，因此密钥的分发与管理是密码学中的重要问题。通常在网络通信中为了保证会话密钥的安全性，要求每一次通信（会话）所使用的会话密钥都不相同，因此如何在每一次会话之前将新的会话密钥传递给通信的对方是一个比较复杂的问题。密钥交换

协议就是在通信双方或多方间协商密钥以进行保密通信的过程，使得参与通信的各方可以获得相同的密钥，但任何其他人都不能获得该密钥。

例 2.7.3（Diffie-Hellman 密钥交换协议） 设 A 和 B 两个用户想协商一个用于保密通信的会话密钥。整个过程可进行如下：首先用户 A 或 B 选定一个素数 p 和有限域 F_p 中的一个本原元 g 并公开。然后：

（1）A 随机选择一个整数 $0 \leqslant x \leqslant p-2$，计算 $X=g^x$ 并发送给 B；

（2）B 随机选择一个整数 $0 \leqslant y \leqslant p-2$，计算 $Y=g^y$ 并发送给 A；

（3）A 计算密钥 $K=Y^x$；

（4）B 计算密钥 $K=X^y$。

至此，用户 A 和 B 得到了一个共同的密钥 g^{xy}。第三者即使知道了 g^x 和 g^y，也不能够计算得到这个会话密钥，除非攻击者能够计算 F_p 中的离散对数问题来得到 x 和 y，因此当 F_p 中的离散对数问题是困难的时，该协议是安全的（注意这句话并不是严密的）。需要注意的是，p 和 g 的选取对于该协议的安全性有着根本的影响，一个必要的条件是 p 应该选取得足够大。

事实上，在例 2.7.3 中，第三者要想求出 A 和 B 共同协商得到的密钥 K，并不一定需要解决离散对数问题，只要他有办法从 g^x 和 g^y 求得 g^{xy} 即可。从 g^x 和 g^y 出发求 g^{xy} 的问题称为计算性 Diffie-Hellman 问题。请看下面的定义。

定义 2.7.9 设 F_q 是一具有 q 个元素的有限域，g 是 F_q 的一个本原元。有限域 F_q 中的：

1）计算性 Diffie-Hellman 问题（CDH）是指：给定了本原元的两个次幂 g^a 和 g^b（注意这里的 a 和 b 并不被计算者所知道），求解 g^{ab}；

2）判定性 Diffie-Hellman 问题（DDH）是指：在给定了本原元的 3 个次幂 g^a、g^b 和 g^c，判定 $g^{ab}=g^c$ 是否成立；

3）间隙性 Diffie-Hellman 问题（Gap-DH）是指：在假定具有解决判定性 Diffie-Hellman 问题的有效方法的情况下，求解计算性 Diffie-Hellman 问题。

很明显，如果在某一特定的有限域 F_q 中，判定性 Diffie-Hellman 问题是困难的话，那么 F_q 中的计算性 Diffie-Hellman 问题和离散对数问题也都是困难的。除了间隙性 Diffie-Hellman 问题外，计算性 Diffie-Hellman 问题和判定性 Diffie-Hellman 问题的困难性都已得到公认，而间隙性 Diffie-Hellman 问题是 2001 年才被正式提出来的。

例 2.7.4（ElGamal 公钥密码系统） 设 p 是一个素数使得有限域 Z_p 中离散对数问题的求解是困难的，$\alpha \in Z_p^*$ 是 Z_p 的一个本原元。秘密选取一个 a 作为秘密密钥保存起来，计算 $\beta=\alpha^a$，并将 p、α、β 公开。

当对一个消息 $x \in Z_p^*$ 进行加密时，首先（秘密）随机选取 $k \in Z_{p-1}$，然后计算 $y_1=\alpha^k \bmod p$，$y_2=x\beta^k \bmod p$，并将 (y_1, y_2) 作为密文。解密时，只需计算 $x=y_2(y_1^a)^{-1} \bmod p$ 即可。

在 ElGamal 公钥密码系统中，明文 x 通过乘上 β^k 而被掩盖起来得到 y_2，α^k 也被作为密文传输。一个人如果知道秘密密钥 a，他就可以从 α^k 计算出 β^k，从而可以从

y_2 中将掩饰去掉而得到明文 x。这个系统的安全性依赖于离散对数问题的困难性，如果能够计算 β 的离散对数，当然可以获得秘密密钥 a，从而可以恢复出明文 x。

2.7.5　线性移位寄存器序列

一个 n 级线性移位寄存器有 n 个寄存器和一个反馈开关电路组成，如图 2.1 所示，从右到左分别称为第 1 级寄存器、第 2 级寄存器、……、第 n 级寄存器。每个寄存器的状态分别用 0 和 1 表示，而 0 和 1 总可看成是有限域 F_2 中的元素。当加上一个移位脉冲时，每级寄存器的内容就移给下一级，最末一级的内容就是输出。为了保持连续工作，将移位寄存器中的某些级的内容在 F_2 中相加之后，反馈到第 1 级去。例如，当第一级的内容为 a_{n-1}，第 2 级的内容为 a_{n-2}，……，第 n 级的内容为 a_0 给定后，可令

$$a_n = c_1 a_{n-1} + c_2 a_{n-2} + \cdots + c_n a_0$$

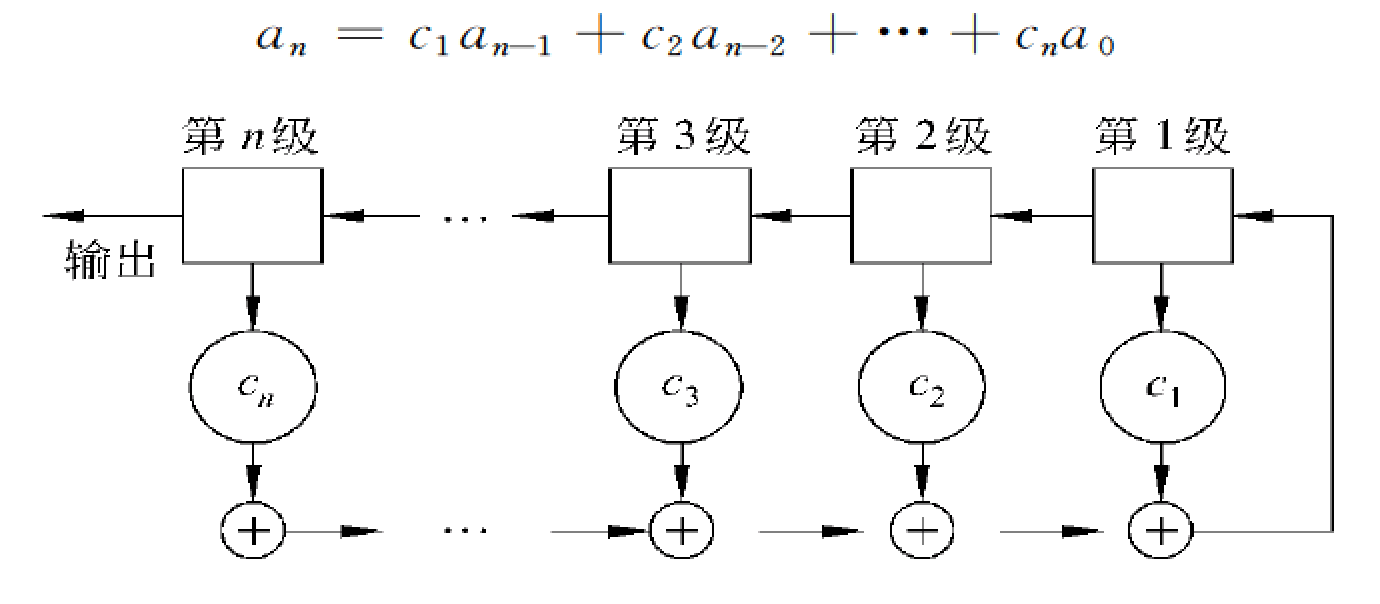


图 2.1　n 级线性移位寄存器

反馈到第 1 级去，其中 $c_1, c_2, \cdots, c_n$ 都是 F_2 中的元素。这样，加上一个移位脉冲后，第 1 级的内容变成

$$a_n = \sum_{i=1}^{n} c_i a_{n-i}$$

第 2 级的内容变成 $a_{n-1}, \cdots$，第 n 级的内容变为 a_1，而输出为 a_0。于是，当线性移位寄存器的 n 个初始值 $a_0, a_1, \cdots, a_{n-1}$ 给定之后，不断地加移位脉冲，n 级线性移位寄存器就会输出一序列

$$a_0, a_1, a_2, \cdots$$

满足线性递推关系(或反馈逻辑)

$$a_k = \sum_{i=1}^{n} c_i a_{k-i} \quad 或 \quad a_k + \sum_{i=1}^{n} c_i a_{k-i} = 0, \quad k \geqslant n \tag{2.2}$$

这个序列就称为(n 级)线性移位寄存器序列或线性递归序列。上述序列之所以称为线性的，是因为它的递推关系式(2.2)，即它的反馈逻辑是线性的。n 级线性移位寄存器序列中任何连续 n 个项都叫做该序列的一个状态，而形如：

$$(a_k, a_{k+1}, \cdots, a_{k+n-1}), \quad k \geqslant 0 \tag{2.3}$$

的状态称为第 k 个状态，记为 s_k。状态 s_0 称为序列的初始状态。

显然，n 级线性移位寄存器序列由它的初始状态 $(a_0, a_1, \cdots, a_{n-1})$ 和它的反馈逻辑完全决定。当 $c_n = 0$，即第 n 级寄存器的内容不参加反馈时，就称这个 n 级线性移

位寄存器是退化的;否则,称为非退化的。非退化的线性移位寄存器产生的序列称为非退化的线性移位寄存器序列。从线性移位寄存器的反馈逻辑构造的如下多项式

$$f(x)=x^n+\sum_{i=1}^{n}c_ix^{n-i};\quad \tilde{f}(x)=1+\sum_{i=1}^{n}c_ix^i$$

分别称为 n 级线性移位寄存器的特征多项式和联接多项式,而把满足递推关系式(2.2)的 n 级线性移位寄存器序列称为由 $f(x)$ 产生的(二元)n 级线性移位寄存器序列或以 $\tilde{f}(x)$ 为联接多项式的 n 级线性移位寄存器序列,简称由 $f(x)$ 产生的序列。通常用符号 $G(f)$ 表示由 $f(x)$ 产生的所有序列的全体组成的集合。显然,$G(f)$ 非空,因为以 0 状态为初始状态的全 0 序列就在 $G(f)$ 中。

一个线性移位寄存器由它的特征多项式,或者联接多项式和它的级数一起完全确定,而且特征多项式和联接多项式为互反多项式。

定理 2.7.11 设 $f(x)=x^n+\sum_{i=1}^{n}c_ix^{n-i}$,那么由 $f(x)$ 产生的序列的总数是 2^n,即 $|G(f)|=2^n$。更进一步,$G(f)$ 可以看成是二元域 F_2 上的 n 维向量空间。

定理 2.7.12 非退化的 n 级线性移位寄存器序列一定是周期序列,而且它的周期不大于 2^n-1。

证明:假设一个非退化的 n 级线性移位寄存器的特征多项式为 $f(x)=x^n+\sum_{i=1}^{n}c_ix^{n-i}$,$a=(a_0,a_1,a_2,a_3,\cdots)$ 是由该线性移位寄存器生成的一个序列,那么 $c_n\neq 0$ 且

$$a_{k+n}=\sum_{i=1}^{n}c_ia_{n+k-i},\quad k\geqslant 0 \tag{2.4}$$

现在考察序列 a 的状态 $s_k(k\geqslant 0)$,它们都是一些 F_2 上的 n 维行向量。注意到 F_2 上不同的 n 维行向量最多只有 2^n 个,因此一定存在 $0\leqslant r<s\leqslant 2^n$,使得 $s_r=s_s$。由于序列的状态和线性递推关系完全决定了从该状态开始之后序列的所有元素,所以对任给的 $k\geqslant r$,$s_{k+s-r}=s_k$。令 $l=s-r$,k_0 是最小的非负整数使得对任给的 $k\geqslant k_0$,$s_{k+l}=s_k$,我们来证 $k_0=0$。

假设 $k_0\geqslant 1$,则根据式(2.4),

$$s_{k_0+n-1+l}=\sum_{i=1}^{n}c_is_{k_0+n-1+l-i}$$

但 $c_n\neq 0$,因此 $c_n=1$,所以

$$s_{k_0-1+l}=s_{k_0+n-1+l}-\sum_{i=1}^{n-1}c_is_{k_0+n-1+l-i}=s_{k_0+n-1}+\sum_{i=1}^{n-1}c_is_{k_0+n-1-i} \tag{2.5}$$

另一方面,根据式(2.4),可以直接得到

$$s_{k_0-1+n}=\sum_{i=1}^{n}c_is_{k_0-1+n-i}$$

再次注意到 $c_n=1$,则有

$$s_{k_0-1}=s_{k_0+n-1}-\sum_{i=1}^{n-1}c_is_{k_0-1+n-i} \tag{2.6}$$

比较式(2.5)和式(2.6)知 $s_{k_0-1}=s_{k_0-1+l}$。这与 k_0 的选取相矛盾。所以 $k_0=0$,即对

所有 $k\geqslant 0, s_{k+l}=s_k$。所以 a 是周期序列。

当 n 级线性移位寄存器序列的周期达到最大值 2^n-1 时，就叫做最长二元 n 级线性移位寄存器序列，简称为 m 序列。m 序列是一类很重要的序列，将在后面讨论。

容易看到，如果 a 是周期为 l 的周期序列，那么 a 显然满足线性递推关系式

$$a_{k+l}+a_k=0, \quad k\geqslant 0$$

因此，周期序列一定是线性移位寄存器序列，其特征多项式为

$$x^l+1$$

更进一步，则有

定理 2.7.13　设 a 是 F_2 上的一个周期序列，$h(x)$、$g(x)$ 是 F_2 上的两个多项式。如果 $a\in G(h)$ 且 $a\in G(g)$，则 $a\in G(h\pm g)$。

证明：设

$$a=(a_0,a_1,a_2,\cdots)$$

$$h(x)=\sum_{i=0}^{n} b_i x^i$$

$$g(x)=\sum_{i=0}^{m} d_i x^i$$

其中 $b_0\neq 0, d_0\neq 0$。那么

$$b_n a_k+b_{n-1}a_{k-1}+\cdots+b_0 a_{k-n}=0, \quad k\geqslant n$$

$$d_m a_k+d_{m-1}a_{k-1}+\cdots+d_0 a_{k-m}=0, \quad k\geqslant m$$

取 $M=\max(m,n)$，并令

$$b_{n+1}=b_{n+2}=\cdots=b_M=0, \quad M>n$$

$$d_{m+1}=d_{m+2}=\cdots=d_M=0, \quad M>m$$

于是有

$$(b_M\pm d_M)a_k+(b_{M-1}\pm d_{M-1})a_{k-1}+\cdots+(b_M\pm d_M)a_{k-M}=0, \quad k\geqslant M$$

由这个递推关系确定的多项式为

$$\sum_{i=0}^{M}(b_i\pm d_i)x^i=h(x)\pm g(x)$$

所以

$$a\in G(h\pm g)$$

定理 2.7.14　设 a 是 F_2 上的一个周期序列，$g(x)$ 是 F_2 上的一个多项式。如果 $a\in G(g)$，则对 F_2 上的任何多项式 $h(x)$，一定有 $a\in G(g\cdot h)$。

证明：

$$a=(a_0,a_1,a_2,\cdots)$$

$$g(x)=\sum_{i=0}^{n} c_i x^{n-i}$$

$$h(x)=\sum_{i=0}^{m} b_i x^{m-i}$$

由定理假设知

$$c_0 a_k + c_1 a_{k-1} + \cdots + c_n a_{k-n} = 0, \quad k \geqslant n$$

取

$$d_0 = c_0, d_1 = c_1, \cdots, d_n = c_n, d_{n+1} = 0$$

那么

$$d_0 a_k + d_1 a_{k-1} + \cdots + d_n a_{k-n} + d_{n+1} a_{k-(n+1)} = 0, \quad k \geqslant n+1$$

这个递推关系式所确定的多项式为

$$\sum_{i=0}^{n+1} d_i x^{n+1-i} = \sum_{i=0}^{n} c_i x^{n+1-i} = xg(x)$$

因此 $a \in G(xg(x))$。从而利用数学归纳法可以证明对任何的 i,$a \in G(x^i g(x))$。所以根据定理 2.7.13,

$$a \in G(\sum_{i=0}^{m} b_i (x^{m-i} g(x))) = G(g(x) \sum_{i=0}^{m} b_i x^{m-i}) = G(g(x)h(x))$$

定理 2.7.15 设 a 是 F_2 上的一个周期序列,那么存在着 F_2 上的一个多项式 $f(x)$ 具有性质:$a \in G(f)$ 且 $a \in G(h(x))$ 当且仅当 $f(x) | h(x)$。更进一步,适合上述性质的多项式 $f(x)$ 是唯一确定的,而且如果 a 是非 0 周期序列,那么 $\deg(f(x)) \geqslant 1$。

证明:令 $S = \{t(x) | a \in G(t(x))\}$,则 S 是一个非空集合,这是因为如果 a 的周期是 l,$x^l + 1$ 一定在 S 中。由定理 2.7.13 和定理 2.7.14 知 S 是 $F_2[x]$ 中的一个理想,因此根据定理 2.2.5,S 是一主理想。设 $f(x)$ 是 S 的一个生成元,则 $h(x) \in S \Leftrightarrow f(x) | h(x)$,因此 $a \in G(h(x)) \Leftrightarrow f(x) | h(x)$。进一步,如果 a 还是非零序列,则非零常数多项式不在 S 中,从而 $\deg(f(x)) \geqslant 1$。定理得证。

定义 2.7.10 设 a 是 F_2 上的一个周期序列,那么根据定理 2.7.15,存在 F_2 上唯一的首一多项式 $f(x)$ 使得 $a \in G(h(x))$ 当且仅当 $f(x) | h(x)$。这个多项式 $f(x)$ 称做 a 的极小多项式。

定理 2.7.16 任给 F_2 上一个多项式 $f(x)$,则必有 F_2 上的一个周期序列存在,它以 $f(x)$ 为极小多项式。

证明:考察 $G(f)$ 中由初始状态 $(\underbrace{0,0,\cdots,0}_{\text{全为0}},1)$ 产生的序列 a。显然 $a \neq 0$。假设 a 的极小多项式是 $h(x) \neq f(x)$,则有 $h(x) | f(x)$,所以 $\deg(h(x)) < \deg(f(x))$。这样 a 将是 $G(h)$ 中从 0 状态得到的序列,因而是 0 序列。这与 $a \neq 0$ 矛盾,所以 a 是以 $f(x)$ 为极小多项式的序列。

定理 2.7.17 设 $f(x)$ 是 F_2 上的一个次数大于 1 的多项式,那么以 $f(x)$ 为极小多项式的线性移位寄存器序列的周期就等于 $f(x)$ 的周期。

证明:设 a 是一以 $f(x)$ 为极小多项式的线性移位寄存器序列,周期为 v,那么 $a \in G(x^v - 1)$。因 $f(x)$ 是 a 的极小多项式,所以 $f(x) | (x^v - 1)$,$p(f) | v$。另一方面,根据多项式周期的定义,有 $f(x) | (x^{p(f)} - 1)$,所以 $a \in G(x^{p(f)} - 1)$,$v | p(f)$。因此 $v = p(f)$。

推论 2.7.3 设 $f(x)$ 是 F_2 上的 n 次多项式,则一个非 0 序列 $a \in G(f)$ 是 n 级 m 序列当且仅当 $f(x)$ 是 n 次本原多项式。

证明:充分性:设 $f(x)$ 是本原多项式,则其周期为 $2^n - 1$。根据定理 2.7.19,

a 的周期为 2^n-1，因而是 m 序列。

必要性：如果 a 是 n 级 m 序列，则 a 的周期为 2^n-1，所以 $G(f)$ 中的每一非 0 序列都以 a 的某一状态为初始状态，因而其周期必为 2^n-1。下面首先证明 $f(x)$ 是不可约多项式。假设 $f(x)$ 可约，$h(x)$ 是它的一个不可约因子，$\deg(h(x))=k<n$，则 $p(h)\leqslant 2^k-1<2^n-1$，因此 $G(h)$ 中的非 0 序列的周期为 $p(h)<2^n-1$。但由 $h(x)\mid f(x)$ 知，$G(h)\subset G(f)$，即 $G(h)$ 中的序列也在 $G(f)$ 中，因此 $G(h)$ 中的非 0 序列的周期也应为 2^n-1。这与 $p(h)<2^n-1$ 矛盾。所以 $f(x)$ 是不可约的。从而 $f(x)$ 的周期应为 2^n-1，所以是本原多项式。

上面讨论了怎样用线性移位寄存器来产生周期尽可能大，证明了在所有不同的 n 级线性移位寄存器中，以 n 次本原多项式为特征多项式的线性移位寄存器产生的序列是 n 级 m 序列，其周期最长。下面将讨论，对于给定的一个有限域 F_q 上的有限序列，如何构造阶数尽可能小的线性移位寄存器来产生它。这个问题称为序列的综合问题。我们把能够产生一个给定序列的阶数最小的线性移位寄存器的阶数称为这个序列的线性复杂度。

设 N 是一个正整数，$a^{(N)}=a_0a_1\cdots a_{N-1}$ 是有限域 F_2 上的一个长度为 N 的序列，$f_N(x)$ 是一个能产生 $a^{(N)}$ 且阶数最小的线性移位寄存器的联接多项式，l_N 是该线性移位寄存器的阶数。我们称二元组 $(f_N(x),l_N)$ 为 $a^{(N)}$ 的线性综合解。应当指出，$\deg(f_N(x))\leqslant l_N$，这是因为产生 $a^{(N)}$ 且阶数最小的线性移位寄存器可能是退化的，这时有 $\deg(f_N(x))<l_N$。另外，约定 0 阶线性移位寄存器的联接多项式为 $f(x)=1$，且长度为 $n(\leqslant N)$ 的零序列 $\underbrace{000\cdots 0}_{n}$ 由 0 阶的线性移位寄存器产生。

已知序列 $a^{(N)}=a_0a_1\cdots a_{N-1}$，求产生 $a^{(N)}$ 并且阶数最小的线性移位寄存器就是求 $a^{(N)}$ 的线性综合解。这一节将要介绍的求序列的线性综合解的 Berlekamp-Massey 算法是一个迭代算法。这个算法是 J. L. Massey 建议的，他指出这个算法就是 E. R. Berlekamp 给出的译 BCH 码时从校验子求找错位多项式的算法。这个算法使用数学归纳法去求一系列的线性移位寄存器

$$(f_n(x),l_n)\quad n=1,2,\cdots,N$$

使得每一个 $(f_n(x),l_n)$ 都是 $a^{(N)}$ 的前 n 项组成的序列的线性综合解。算法的具体描述如下。

算法 2.7.2（Berlekamp-Massey 算法）

(1) 设 n_0 是一个非负整数，满足

$$a_0=a_1=\cdots=a_{n_0-1}=0,\quad a_{n_0}\neq 0$$

则取

$$d_0=d_1=\cdots=d_{n_0-1}=0,\quad d_{n_0}=a_{n_0}$$

并令

$$f_1(x)=f_2(x)=\cdots=f_{n_0}(x)=1$$

$$l_1=l_2=\cdots=l_{n_0}=0$$

同时可以取任意一个 n_0+1 级的线性移位寄存器作为 $(f_{n_0+1}(x),l_{n_0+1})$，但为了确定

起见，令

$$f_{n_0+1}(x) = 1 - d_{n_0} x^{n_0+1}, \quad l_{n_0+1} = n_0 + 1$$

（2）假设$(f_i(x), l_i)$，$1 \leqslant i \leqslant n \leqslant N$ 已经求得。而

$$l_1 = l_2 = \cdots = l_{n_0} < l_{n_0+1} \leqslant l_{n_0+1} \leqslant \cdots \leqslant l_n$$

令

$$f_n(x) = 1 + c_{n,1}x + \cdots + c_{n,l_n} x^{l_n}$$

并计算

$$d_n = a_n + c_{n,1}a_{n-1} + \cdots + c_{n,l_n} a_{n-l_n}$$

如果 $d_n = 0$，则取

$$f_{n+1}(x) = f_n(x), \quad l_{n+1} = l_n$$

如果 $d_n \neq 0$，这时一定存在 $1 \leqslant m < n$，使

$$l_m < l_{m+1} = l_{m+2} = \cdots = l_n$$

取

$$f_{n+1}(x) = f_n(x) - d_n d_m^{-1} x^{n-m} f_m(x)$$
$$l_{n+1} = \max\{l_n, n+1-l_n\}$$

例 2.7.5 求周期为 8 的序列 $a^{(8)} = 00101101$ 的线性综合解。在这里 $a_0 = 0$，$a_1 = 0$，$a_2 = 1$，$a_3 = 0$，$a_4 = 1$，$a_5 = 1$，$a_6 = 0$，$a_7 = 1$。

首先 $n_0 = 2$，因此

$$d_0 = d_1 = 0, \quad d_2 = 1$$
$$f_1(x) = f_2(x) = 1, \quad f_3(x) = 1 - x^3$$
$$l_1 = l_2 = 0, \quad l_3 = 3$$

计算 $d_3 = a_3 - a_0 = 0 + 0 = 0$，因此取

$$f_4(x) = f_3(x) = 1 - x^3$$
$$l_4 = l_3 = 3$$

计算 $d_4 = a_4 - a_1 = 1 - 0 = 1 \neq 0$，这时 $l_2 < l_3 = l_4$，因此 $m = 2$，

$$f_5(x) = f_4(x) - d_4 d_2^{-1} x^{4-2} f_2(x) = 1 - x^3 - x^2$$
$$= 1 + x^2 + x^3$$
$$l_5 = \max\{l_4, 4 + 1 - l_4\} = 3$$

计算 $d_5 = a_5 + a_3 + a_2 = 1 + 0 + 1 = 0$，因此

$$f_6(x) = f_4(x) = 1 + x^2 + x^3$$
$$l_6 = l_5 = 3$$

计算 $d_6 = a_6 + a_4 + a_3 = 0 + 1 + 0 = 1 \neq 0$，这时 $l_2 < l_3 = l_4 = l_5$，因此 $m = 2$，

$$f_7(x) = f_6(x) - d_6 d_2^{-1} x^{6-2} f_2(x) = 1 + x^2 + x^3 - x^4$$
$$= 1 + x^2 + x^3 + x^4$$
$$l_7 = \max\{l_6, 7 - l_6\} = 4$$

计算 $d_7 = a_7 + a_5 + a_4 + a_3 = 1 + 1 + 1 + 0 = 1 \neq 0$，这时 $l_6 < l_7$，因此 $m = 6$，

$$f_8(x) = f_7(x) - d_7 d_6^{-1} x^{7-6} f_6(x) = 1 + x^2 + x^3 + x^4 - x(1 + x^2 + x^3)$$

$$= 1 + x + x^2$$

$$l_8 = \max\{l_7, 8 - l_7\} = 4$$

所以 $a^{(8)}=00101101$ 的线性综合解为$(1+x+x^2, 4)$。

2.8　注记

本章较为全面地介绍了信息安全中所用到的一些代数学基础知识，但由于篇幅限制，相关的介绍还相当简略。有关代数学较为详尽的知识，读者可参看文献[2]、[7]和[8]，有关有限域和线性移位寄存器序列较为详尽的内容可参看文献[1]、[3]、[4]和[9]，有关 Gröbner 基的理论，读者可查看文献[5]和[10]，关于 Ritt-吴特征列方法，读者可参看文献[6]。

参考文献

[1]　林东岱. 代数学基础与有限域. 北京：高等教育出版社，2006

[2]　Zhe-xian Wan. Lectures on Finite Fields and Galois Rings, World Scientific, 2003

[3]　万哲先. 代数与编码. 第三版. 北京：高等教育出版社，2007

[4]　冯克勤. 有限域. 长沙：湖南教育出版社，1991

[5]　刘木兰. Gröbner 基理论及其应用. 北京：科学出版社，2000

[6]　吴文俊. 数学机械化. 北京：科学出版社，2003

[7]　Nathan Jacobson, Basic Algebra (I, II). W. H. Freeman and Company, 1974

[8]　Thomas W. Hungerford. Algebra. Springer Verlag, 1974

[9]　Rudolf Lidl, Harald Niederreiter, Cohn P M. Finite Field. Addison-Wesley Publishing Company, 1983

[10]　Thomas Becker, Volker Weispfenning. Gröbner Bases. Springer Verlag, 1993

第3章　椭圆曲线方法与技术

椭圆曲线应用于密码学开始于 Koblitz 和 Miller。这两位学者几乎同时提出了椭圆曲线密码体制的概念。椭圆曲线密码体制的安全性基于椭圆曲线的 Mordell-Weil 群上离散对数的计算困难性。

椭圆曲线密码体制有许多优点。首先是密钥短，密钥长度为 106 比特的椭圆曲线密码体制的安全强度相当于密钥长度为 512 比特的 RSA 密码体制的安全强度；其次是计算速度快，密码学中所用的椭圆曲线是定义在有限域上的代数曲线，因此它有代数和几何两方面的性质。在椭圆曲线上，点的逆元素容易计算，同时椭圆曲线上的加法群有模结构，因此可供选择的算法就比较多，计算速度也比较快；还有就是定义在同一个有限域上的椭圆曲线有许多条，因此需更换密码体制时只要更换一条定义在同一个有限域上的椭圆曲线，从而有限域的算法还可以继续使用。所以与基于有限域上离散对数问题的密码体制相比，椭圆曲线密码体制中涉及有限域算法的芯片可以重复使用，也就是说通用性比较好。

由于以上这些特点，椭圆曲线密码体制特别适合于应用到计算资源有限的环境中。另外，椭圆曲线上可以定义双线性对（Weil 对和 Tate 对，称之为“对子”），对子把椭圆曲线上离散对数问题转化为有限域上离散对数问题，这种优秀的密码学性质，被用来构造新型的密码体制。

本章主要围绕椭圆曲线在信息安全中的应用，将介绍椭圆曲线的一些基本概念和基本原理。最后，作为这些椭圆曲线理论的应用，还将介绍一些典型的密码体制。

3.1　基本概念

本节重点介绍一些椭圆曲线的基本概念。

3.1.1　椭圆曲线的定义

设 K 是一个域（为了便于理解，不妨把 K 看成实数域 R，但是在密码学应用中，K 一般是有限域）。从平面解析几何的角度来说，定义在 K 上的椭圆曲线 E 是一条由 Weierstrass 方程

$$y^2+a_1xy+a_3y=x^3+a_2x^2+a_4x+a_6,\quad a_i\in K, i=1,2,3,4,6 \tag{3.1}$$

定义的非奇异（即处处光滑，或者不严格地说自己和自己没有交点）的 3 次曲线，再“人为地”添加一个无穷远点（用∞表示）所得到的曲线，记为 E/K。有时说起由方程(3.1)定义的椭圆曲线时，并不特意指出包含无穷远点，但都默认它自然包含了一个无穷远点。

由此可见，椭圆曲线 E/K 是 $K\times K$ 平面上由方程(3.1)的全体解构成的图形。

设 $L \supset K$ 是 K 的扩域，方程(3.1)在 $L \times L$ 平面上的解称为椭圆曲线 E 的 L 有理点。E 的全体 L 有理点记为 $E(L)$。根据椭圆曲线的定义，无穷远点必须包含在 $E(L)$ 内：

$$
\begin{aligned}
E(L) = \{(x,y) \in L \times L \mid y^2 + a_1xy + a_3y \\
= x^3 + a_2x^2 + a_4x + a_6\} \cup \{\infty\}
\end{aligned}
$$

例 3.1.1　两条椭圆曲线如图 3.1 所示。

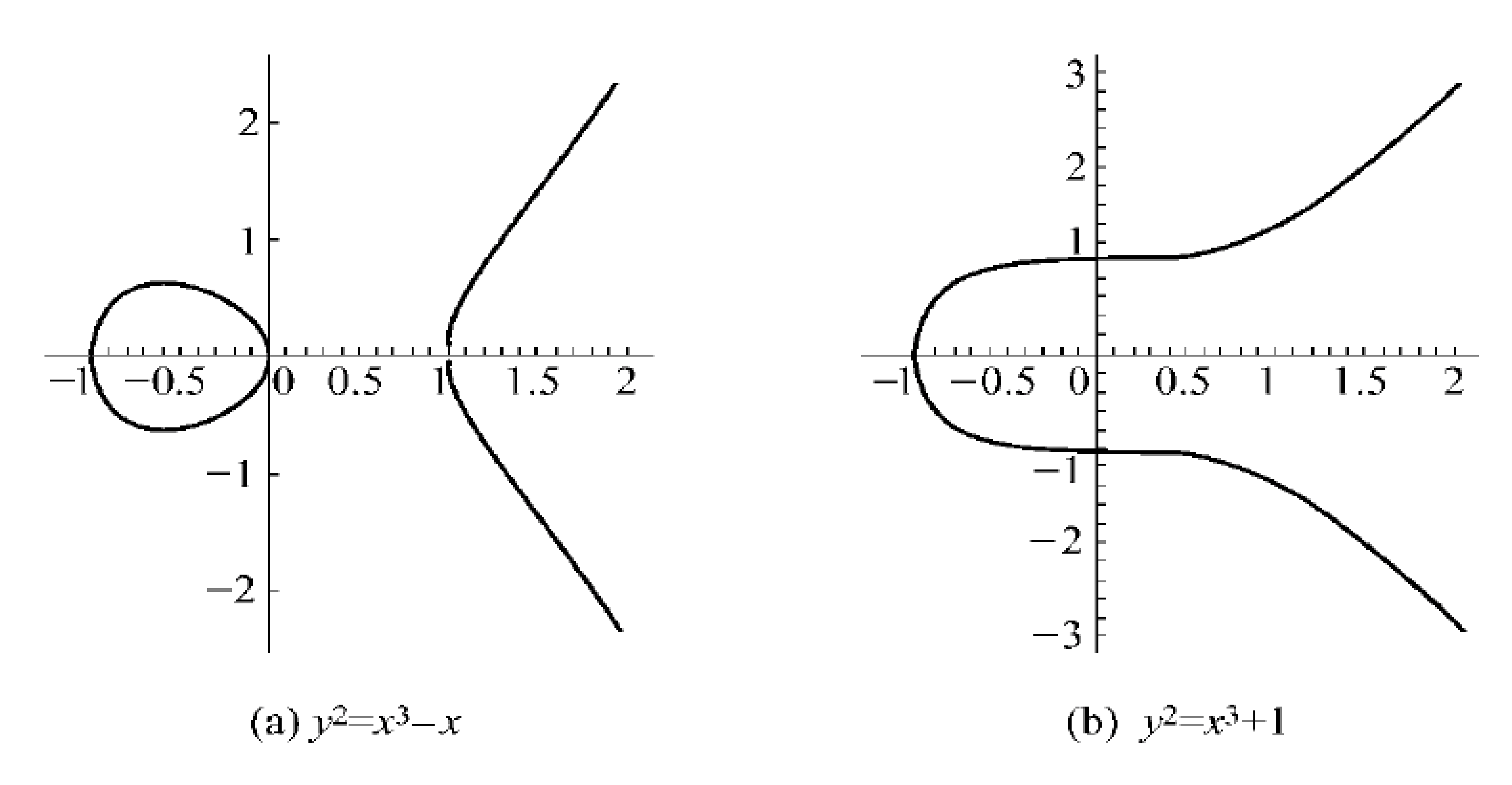


图 3.1　两条椭圆曲线

设 E 是一条由 Weierstrass 方程(3.1)定义的曲线，则可以定义以下参数：

$$
\begin{aligned}
b_2 &= a_1^2 + 4a_2 \\
b_4 &= 2a_4 + a_1a_3 \\
b_6 &= a_3^2 + 4a_6 \\
b_8 &= a_1^2a_6 + 4a_2a_6 - a_1a_3a_4 + a_2a_3^2 - a_4^2 \\
c_4 &= b_2^2 - 24b_4 \\
\Delta &= -b_2^2b_8 - 8b_4^3 - 27b_6^2 + 9b_2b_4b_6 \\
j(E) &= c_4^3/\Delta
\end{aligned}
$$

其中 Δ 称为这个 Weierstrass 方程的判别式。如果 $\Delta \neq 0$，则 $j(E)$ 有定义，称为这个方程的 j 不变量。判别式和 j 不变量是椭圆曲线的重要参数。

定理 3.1.1　由方程(3.1)定义的曲线是一条非奇异曲线的充分必要条件是 $\Delta \neq 0$。

证明：把方程(3.1)写成形如 $F(x,y)=0$ 的隐函数形式。根据多元微积分中的定理可知，由 $F(x,y)=0$ 定义的曲线是非奇异曲线当且仅当方程组 $\begin{cases} F(x,y)=0 \\ F_x(x,y)=0 \\ F_y(x,y)=0 \end{cases}$ 没有解；用 Δ 的定义直接验证可得定理。

定理 3.1.2　定义在 K 上的两条椭圆曲线 E_1/K 和 E_2/K 同构，则 $j(E_1)=j(E_2)$；如果 K 是一个代数封闭域，则由 $j(E_1)=j(E_2)$ 也可以得到 E_1/K 和 E_2/K 同构。

这里同构的意思是两条曲线之间的方程可以通过一个有理变换互相转化，而且这个有理变换把无穷远点变到无穷远点。

例 3.1.2 因为判别式 $\Delta=0$，所以由图 3.2 所示的两条曲线是奇异曲线，因而不是椭圆曲线：

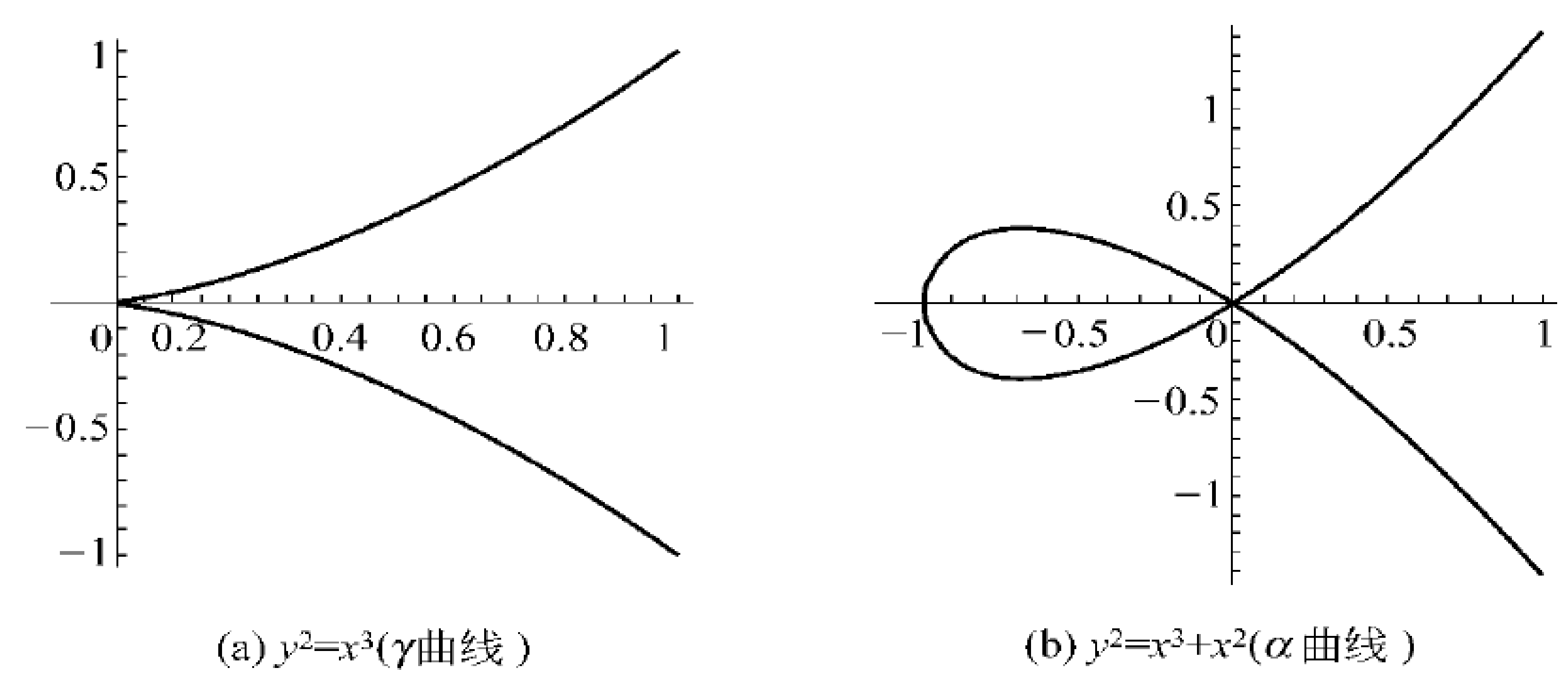


图 3.2 两条奇异曲线

椭圆曲线的 Weierstrass 方程在形式上是复杂的，但总可以通过坐标变换把它变成相对比较简单的形式，这是一个标准过程，请参考文献[4]。

（1）$\mathrm{char}(K)\neq 2,3$，则 E/K 的方程可以化成以下形式：

$$E: y^2 = x^3 + ax + b \quad a,b \in K \tag{3.2}$$

（2）$\mathrm{char}(K)=2$，则 E/K 的方程可以化成以下两种形式：

① $j(E)\neq 0$，E：$y^2+xy=x^3+a_2x^2+a_6 \quad a_2,a_6\in K$ （3.3）

② $j(E)=0$，E：$y^2+a_3y=x^3+a_4x+a_6 \quad a_3,a_4,a_6\in K$ （3.4）

（3）$\mathrm{char}(K)=3$，则 E/K 的方程可以化成以下两种形式：

① $j(E)\neq 0$，E：$y^2=x^3+a_2x^2+a_6 \quad a_2,a_6\in K$ （3.5）

② $j(E)=0$，E：$y^2=x^3+a_4x+a_6 \quad a_4,a_6\in K$ （3.6）

3.1.2 椭圆曲线上的 Mordell-Weil 群

椭圆曲线上的点在“弦切律”下构成一个群。

定义 3.1.1 椭圆曲线上点的加法（弦切律） 如图 3.3(a)、(b)所示，设 P、$Q\in E$，ℓ 是过 P、Q 两点的直线（如果 $P=Q$，则取 ℓ 为过 P 点的切线），则由于椭圆曲线是一条 3 次曲线，因此 ℓ 必和 E 相交于第三点 R。令 ℓ' 为过 R 和无穷远点的直线（即过 R 且与 x 轴垂直的直线），则 ℓ' 与 E 的第三个交点即为 P 与 Q 的“和”记为 $P+Q$。

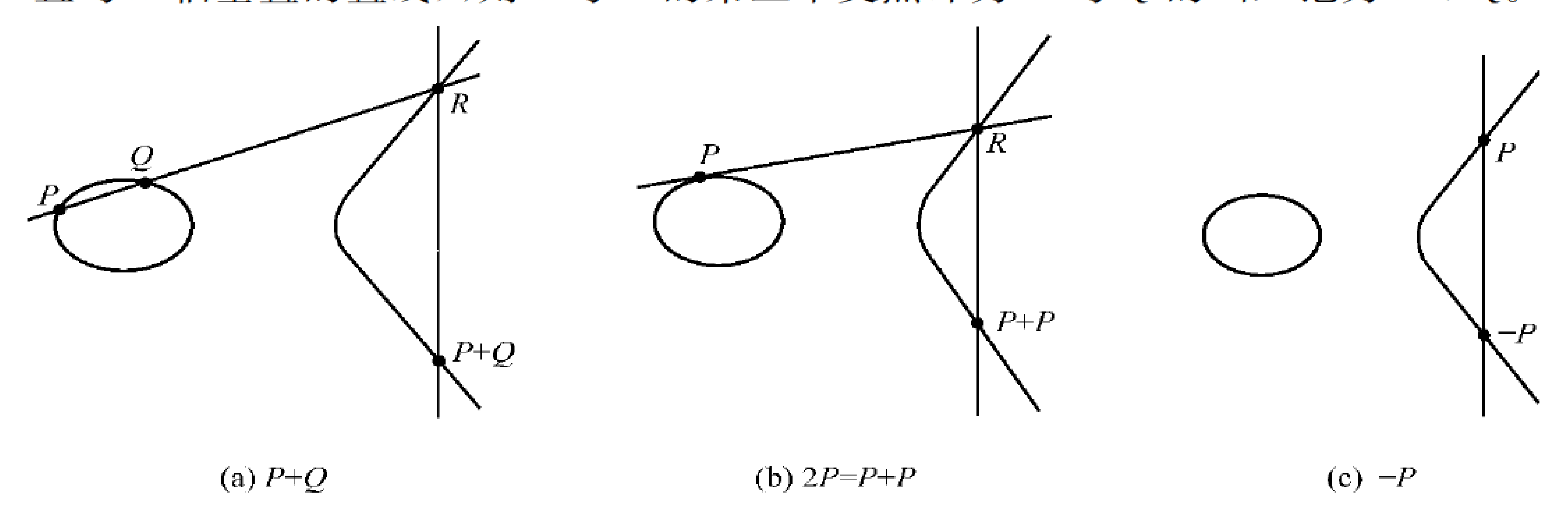


图 3.3 实数域上椭圆曲线弦切律的几何表示

P 点的逆 $-P$ 即为过 P 且与 x 轴垂直的直线与曲线的交点，如图 3.3(c)所示。

图 3.3 所示为实数域上椭圆曲线的弦切律的几何表示，如果要考虑其他域上的椭圆曲线的弦切律，也可以甚至必须借助这些图形思考问题。

由于在平面解析几何的范围内考虑椭圆曲线问题，因此无穷远点在上面的图中表示不出来，但理解加法时要默认它是存在的。比如在定义中垂直于 x 轴的直线 ℓ' 和曲线的交点在图中只显示出有两个，但是理论上 3 次曲线和直线的交点应该有 3 个，那么第三个点就是在无穷远点。这从另外一个方面说明，在定义椭圆曲线时"人为"引入的那个无穷远点是实际存在的，在引入射影几何后，这个无穷远点就自然出现了。

还有一点需要注意的是，如果 E/K 是一条定义在 K 上的椭圆曲线，L 是 K 的扩域，P、Q 是 E 的 L-有理点，则过 P、Q 的直线和 E 的第三个交点一定是 E 的 L 有理点。这是因为直线和曲线的联立方程组的系数都取自域 L，而方程组有 3 个解，其中两个解就是 P 和 Q，那么第三个解一定是 L-有理点。

定理 3.1.3　设 E/K 是一条定义在域 K 上的椭圆曲线，$L \supset K$ 是 K 的扩域，则 E 的全体有理点 $E(L)$ 在弦切律下构成一个交换群，其中单位元就是无穷远点，即用弦切律定义的加法符合以下规律：

(1) 结合律　$(P+Q)+R=P+(Q+R), \forall P,Q,R \in E(L)$；

(2) 存在零元素　$P+\infty=\infty+P, \forall P \in E(L)$；

(3) 存在逆元素　$P+(-P)=(-P)+P, \forall P \in E(L)$；

(4) 交换律　$P+Q=Q+Q, \forall P,Q \in E(L)$。

定理的证明除第一条结合律外都是平凡的。利用代数几何理论给出的结合律的证明可以在文献[4]中找到；也可以通过弦切律的定义直接进行验证，但是那样的话计算量会比较大，而且需要有耐心才可以；Knapp 在文献[3]中给出了一个只利用初等解析几何和线性代数的证明，有兴趣的读者可以参考。

需要指出的是，以上用弦切律定义的群被称为 Mordell-Weil 群。其实发现这是个群结构的时间比 Mordell 和 Weil 要早，可以追溯到 Poincare 甚至 Abel。这个群之所以被称为 Mordell-Weil 群，是因为 Mordell 最早给出了有理数域上椭圆曲线 Mordell-Weil 群的有限生成定理，而 Weil 推广了他的结果。

对于密码学工作者来说，工作的域一般是有限域，在某些情况下也会涉及一些 p-adic 域和复数域。有限域上椭圆曲线和 p-adic 域上椭圆曲线没有直观的图示，复数域上椭圆曲线的图形是个三维空间中的环面。在这些情况下思考问题会缺乏直观的感觉，此时最好借用实数域上椭圆曲线的图形来支持我们的直觉。

点加的表达式：设 E 是由方程(3.1)定义的一条椭圆曲线，则由弦切律定义的加法的表达式可以通过平面解析几何中求曲线和直线的交点的方法推导出来。这个推导的工作量不大，难度也小，把详细推导过程留给读者，这里只是把结果列出来。

(1) 逆元素：$P=(x,y) \in E$，则 $-P=(x,-y-a_1x-a_3)$。

(2) 加法：记 $P_3=P_1+P_2$，其中 $P_i=(x_i,y_i)$，$i=1,2,3$。

1) 若 $P_1=-P_2$，即 $x_1=x_2$，$y_1+y_2+a_1x_1+a_3=0$，则 $P_1+P_2=\infty$。

2）若 $P_1 \neq -P_2$，令

$$\lambda = \begin{cases} \dfrac{y_2 - y_1}{x_2 - x_1} & x_1 \neq x_2 \\ \dfrac{3x_1^2 + 2a_2x_1 + a_4 - a_1y_1}{2y_1 + a_1x_1 + a_3} & x_1 = x_2 \end{cases}$$

则 P_3 的坐标由下式给出：

$$\begin{cases} x_3 = \lambda^2 + a_1\lambda - a_2 - x_1 - x_2 \\ y_3 = \lambda(x_1 - x_3) - y_1 - a_1x_3 - a_3. \end{cases}$$

如果采取曲线方程的简化形式(3.6)，则以上加法的表达式有更简单的形式，这在实践中有重要的作用。

例 3.1.3 椭圆曲线 E：$y^2 = x^3 + ax + b$，$(a, b \in K, \mathrm{char}(K) \neq 2, 3)$上点加的表达式。

(1) 逆元素：设 $P = (x, y)$，则 $-P = (x, -y)$。

(2) 一般加法：设 $P_3 = P_1 + P_2$，其中 $P_i = (x_i, y_i)$，$i = 1, 2, 3$。

1）若 $P_1 = -P_2$，即 $x_1 = x_2$，$y_1 = -y_2$，则 $P_1 + P_2 = \infty$。

2）若 $P_1 \neq -P_2$，令

$$\lambda = \begin{cases} \dfrac{y_2 - y_1}{x_2 - x_1} & x_1 \neq x_2 \\ \dfrac{3x_1^2 + a}{2y_1} & x_1 = x_2 \end{cases}$$

则 P_3 的坐标由下式给出：

$$\begin{cases} x_3 = \lambda^2 - x_1 - x_2 \\ y_3 = \lambda(x_1 - x_3) - y_1 \end{cases}$$

例 3.1.4 椭圆曲线 E：$y^2 + xy = x^3 + a_2x^2 + a_6$，$(a_2, a_6 \in K, \mathrm{char}(K) = 2)$上点加的表达式。

(1) 逆元素：设 $P = (x, y)$，则 $-P = (x, y + x)$。

(2) 一般加法：设 $P_3 = P_1 + P_2$，其中 $P_i = (x_i, y_i)$，$i = 1, 2, 3$。

1）若 $P_1 = -P_2$，即 $x_1 = x_2$，$y_1 = y_2 + x_2$，则 $P_1 + P_2 = \infty$。

2）若 $P_1 \neq -P_2$，且 $P_1 \neq P_2$ 时，P_3 的坐标由下式给出：

$$\begin{cases} x_3 = \left(\dfrac{y_1 + y_2}{x_1 + x_2}\right)^2 + \dfrac{y_1 + y_2}{x_1 + x_2} + x_1 + x_2 + a_2 \\ y_3 = \left(\dfrac{y_1 + y_2}{x_1 + x_2}\right)(x_1 + x_3) + x_3 + y_1. \end{cases}$$

3）若 $P_1 \neq -P_2$，且 $P_1 = P_2$ 时，P_3 的坐标由下式给出：

$$\begin{cases} x_3 = x_1^2 + \dfrac{a_6}{x_1^2} \\ y_3 = x_1^2 + \left(x_1 + \dfrac{y_1}{x_1}\right)x_3 + x_3 \end{cases}$$

例 3.1.5 椭圆曲线 E：$y^2 + a_3y = x^3 + a_4x + a_6$，$(a_2, a_3, a_6 \in K, \mathrm{char}(K) = 2)$上

点加的表达式。

(1) 逆元素：设 $P=(x,y)$，则 $-P=(x,y+a_3)$。

(2) 一般加法：设 $P_3=P_1+P_2$，其中 $P_i=(x_i,y_i)$，$i=1,2,3$。

1) 若 $P_1=-P_2$，即 $x_1=x_2$，$y_1=y_2+a_3$，则 $P_1+P_2=\infty$。

2) 若 $P_1\neq -P_2$，且 $P_1\neq P_2$ 时，P_3 的坐标由下式给出：

$$\begin{cases} x_3 = \left(\dfrac{y_1+y_2}{x_1+x_2}\right)^2 + x_1 + x_2 \\ y_3 = \left(\dfrac{y_1+y_2}{x_1+x_2}\right)(x_1+x_3) + y_1 + a_3. \end{cases}$$

3) 若 $P_1\neq -P_2$，且 $P_1=P_2$ 时，P_3 的坐标由下式给出：

$$\begin{cases} x_3 = \dfrac{x_1^4+a_4^2}{a_3^2} \\ y_3 = \left(\dfrac{x_1^2+a_4}{a_3}\right)(x_1+x_3) + y_1 + a_3 \end{cases}$$

特征 3 的情形请读者自己补齐或参见文献[1,4]。

3.2　射影坐标和 Jacobi 坐标

椭圆曲线上的点有几种不同的坐标表示，此前使用的平面解析几何中的坐标平面被称为仿射平面，所以仿射坐标平面上的点的坐标表示也称为仿射坐标，通常记为 (x,y)，而无穷远点没有相应的坐标表示，用 ∞ 或者 O 表示。

用仿射坐标表示椭圆曲线上点在做点加时会涉及有限域的除法，而除法在计算中要消耗较多的计算时间。在椭圆曲线密码体制的应用中，每一次计算都要涉及数百次点加运算，所以能够在做点加时省下一些除法运算，那对于提高系统的效率是有重要意义的。这个时候需要射影坐标和 Jacobi 坐标。

3.2.1　射影坐标

我们把射影坐标理解为仿射坐标通分后把公分母用另一个分量表示的一种坐标表示方法。这样，给射影坐标乘上一个非零常数，那这个坐标和原来的坐标表示的是同一个点。

点 (x,y) 的坐标"通分"以后写为 $(X/Z,Y/Z)$，把公分母写成另一个分量，则这个点的表示为 $[X,Y,Z]$。这里用方括号表示坐标 $[X,Y,Z]$ 和 $[\lambda X,\lambda Y,\lambda Z]$ 当 λ 不等于零时表示同一个点。这是因为，把 $[X,Y,Z]$ 和 $[\lambda X,\lambda Y,\lambda Z]$ 变成仿射坐标就是 $(X/Z,Y/Z)$ 和 $(\lambda X/\lambda Z,\lambda Y/\lambda Z)$，这是同一个点的坐标。

射影平面是在仿射平面的基础上加上一些无穷远点构成的。动用一点灵活性，减少一点严格性，不妨称仿射平面上的点为"有穷远点"，以示对无穷远点的区别。射影平面上的有穷远点的射影坐标 $[X,Y,Z]$ 的特点是 $Z\neq 0$；无穷远点的射影坐标形如 $[X,Y,0]$，其中 X 和 Y 中至少有一个不为零。下面就来计算一下椭圆曲线上无穷远点的射影坐标。

设 E：$y^2+a_1xy+a_3y=x^3+a_2x^2+a_4x+a_6(a_i\in K)$是一条椭圆曲线，如果 $P=(x,y)$是 E 上一个点，则把 P 的坐标代入上述方程可使方程两边相等。设 P 的射影坐标为 $P=[X,Y,Z]$，则当 $Z\neq0$ 时有

$$x = X/Z, \quad y = Y/Z$$

代入上述椭圆曲线的方程有

$$\left(\frac{Y}{Z}\right)^2 + a_1\frac{XY}{Z^2} + a_3\frac{Y}{Z} = \left(\frac{X}{Z}\right)^3 + a_2\left(\frac{X}{Z}\right)^2 + a_4\frac{X}{Z} + a_6$$

方程两边各项通分后乘上 Z^3 得

$$Y^2Z + a_1XYZ + a_3YZ^2 = X^3 + a_2X^2Z + a_4XZ^2 + a_6Z^3$$

这个方程称为椭圆曲线的齐次方程。因为无穷远点的坐标的特点是 $Z=0$，把它代入齐次方程后发现方程变成

$$X^3 = 0$$

也就是说，椭圆曲线上无穷远点的坐标一定是$[0,Y,0]$的样子，而点的射影坐标的 3 个分量一定要有一个不为零，因此 $Y\neq0$。所以椭圆曲线上的无穷远点一定是$[0,1,0]$。

可以看到，用仿射坐标表示椭圆曲线上的点，无穷远点就没有坐标表示；而在射影坐标的情形，无穷远点有一个自然的坐标表示。这说明射影平面比仿射平面更具完备性。

3.2.2 Jacobi 坐标

Jacobi 坐标用$[X,Y,Z]$表示点的坐标。Jacobi 坐标和仿射坐标(x,y)的关系为

$$x = X/Z^2, \quad y = Y/Z^3$$

当域的特征大于 3 时，方程的形式为

$$Y^2 = X^3 + aXZ^4 + bZ^6$$

当域的特征是 2 时，方程的形式分别是

$$Y^2 + XYZ = X^3 + a_2X^2Z^2 + a_6Z^6, \quad j \neq 0$$

$$Y^2 + a_3YZ^3 = X^3 + a_4XZ^4 + a_6Z^6, \quad j = 0$$

在 Jacobi 坐标下，无穷远点的坐标是$[0,1,0]$。仿射坐标(x,y)到 Jacobi 坐标的转换为 $X=x,Y=y,Z=1$。

使用 Jacobi 坐标的点加算法比采取射影坐标更快一些。以下就给出两条曲线的点加运算的公式。

定义在特征大于 3 的曲线 $Y^2=X^3+aXZ^4+bZ^6$ 的点加公式。其中 $P_i=(X_i,Y_i,Z_i)$，$i=1,2,3$，符合 $P_3=P_1+P_2$。

(1) $P_1\neq P_2$：

$$\lambda_1 = X_1Z_2^2$$

$$\lambda_2 = X_2Z_1^2$$

$$\lambda_3 = \lambda_1 - \lambda_2$$

$$\lambda_4 = Y_1Z_2^3$$

$$
\begin{aligned}
\lambda_5 &= Y_2 Z_1^3 \\
\lambda_6 &= \lambda_4 - \lambda_5 \\
\lambda_7 &= \lambda_1 + \lambda_2 \\
\lambda_8 &= \lambda_4 + \lambda_5 \\
Z_3 &= Z_1 Z_2 \lambda_3 \\
X_3 &= \lambda_6^2 - \lambda_7 \lambda_3^2 \\
\lambda_9 &= \lambda_7 \lambda_3^2 - 2X_3 \\
Y_3 &= (\lambda_9 \lambda_6 - \lambda_8 \lambda_3^3)/2
\end{aligned}
$$

（2）$P_1 = P_2$：

$$
\begin{aligned}
\lambda_1 &= 3X_1^2 + aZ_1^4 \\
Z_3 &= 2Y_1 Z_1 \\
\lambda_2 &= 4X_1 Y_1^2 \\
X_3 &= \lambda_1^2 - 2\lambda_2 \\
\lambda_3 &= 8Y_1^4 \\
Y_3 &= \lambda_1(\lambda_2 - \lambda_3) - \lambda_3
\end{aligned}
$$

定义在特征 2 的曲线 $Y^2 + XYZ = X^3 + a_2 X^2 Z^2 + a_6 Z^6$ 的点加公式。其中 $P_i = (X_i, Y_i, Z_i)$，$i = 1, 2, 3$，符合 $P_3 = P_1 + P_2$。

（1）$P_1 \neq P_2$：

$$
\begin{aligned}
\lambda_1 &= X_1 Z_2^2 \\
\lambda_2 &= X_2 Z_1^2 \\
\lambda_3 &= \lambda_1 + \lambda_2 \\
\lambda_4 &= Y_1 Z_2^3 \\
\lambda_5 &= Y_2 Z_1^3 \\
\lambda_6 &= \lambda_4 + \lambda_5 \\
\lambda_7 &= Z_1 \lambda_3 \\
\lambda_8 &= \lambda_6 X_2 + \lambda_7 Y_2 \\
Z_3 &= \lambda_7 Z_2 \\
\lambda_9 &= \lambda_9 + Z_3 \\
X_3 &= a_2 Z_3^2 + \lambda_6 \lambda_9 + \lambda_3^3 \\
Y_3 &= \lambda_9 X_3 + \lambda_8 \lambda_7^2
\end{aligned}
$$

（2）$P_1 = P_2$：

$$
\begin{aligned}
Z_3 &= X_1 Z_2^2 \\
X_3 &= (X_1 + a_6 Z_1^2)^4 \\
\lambda &= Z_3 + X_1^2 + Y_1 Z_1 \\
Y_3 &= X_1^4 Z_3 + \lambda X_3
\end{aligned}
$$

3.3 自同态

本节讨论椭圆曲线 E 上的自同态。

定义 3.3.1 有理映射 $\alpha: E(\overline{K})\to E(\overline{K})$ 称为椭圆曲线 E 的自同态，如果它符合

$$\alpha(P_1+P_2)=\alpha(P_1)+\alpha(P_2)$$

这里 α 是有理映射的意思是，存在有理函数（多项式的商）$R_1(x,y)$，$R_2(x,y)$，使得对任意的$(x,y)\in E(\overline{K})$，都有 $\alpha(x,y)=(R_1(x,y),R_2(x,y))$。

例 3.3.1 设 $E: y^2=x^3+ax+b$ 是一条椭圆曲线，则 $P\mapsto nP$ 是 E 的自同态，因此其形式为 $n(x,y)=(R_1(x,y),R_2(x,y))$。特别地，$n=2$ 时有

$$2(x,y)=\left(\left(\frac{3x^2+a}{2y}\right)^2-2x,\left(\frac{3x^2+a}{2y}\right)\left(3x-\left(\frac{3x^2+a}{2y}\right)^2\right)-y\right)$$

为了简单起见，考虑特征不等于 2,3 的域上定义的椭圆曲线 E：$y^2=x^3+ax+b$。此时对任意的$(x,y)\in E(\overline{K})$都有 $y^2=x^3+ax+b$，因此 E 上的有理函数都可以写成以下形式：

$$R(x,y)=\frac{p_1(x)+p_2(x)y}{p_3(x)+p_4(x)y}$$

进而用 $p_3(x)-p_4(x)y$ 对上式的分母“有理化”后有

$$R(x,y)=\frac{q_1(x)+q_2(x)y}{q_3(x)}$$

又因为

$$\alpha(x,-y)=\alpha(-(x,y))=-\alpha(x,y)$$

故而有

$$R_1(x,-y)=R_1(x,y),\quad R_2(x,-y)=-R_2(x,y)$$

因此有

$$\alpha(x,y)=(r_1(x),r_2(x)y)$$

其中 $r_1(x)$、$r_2(x)$是 x 的有理函数。

记 $r_1(x)=p(x)/q(x)$，其中$(p(x),q(x))=1$。如果在点(x,y)处有 $q(x)=0$，则令 $\alpha(x,y)=\infty$；如果 $q(x)\neq 0$，则可以证明（留给读者证明）$r_2(x)\neq 0$，因此 α 的定义是良好的。如果 α 是一个非平凡自同态，则定义 α 的次数为

$$\deg(\alpha)=\max\{\deg p(x),\deg q(x)\}$$

规定 $\deg(0)=0$。如果$\frac{\mathrm{d}}{\mathrm{d}x}r_1(x)\neq 0$，则称 α 是可分的；否则称 α 为不可分的。

关于椭圆曲线自同态的一个基本定理表明，椭圆曲线上的全体自同态构成一个环，并且同构于下列 3 种环之一[1]：

(1) 整数环；

(2) 虚二次代数数域的子环；

(3) 四元数域的子环。

因此，椭圆曲线 E 的自同态映射 α 在 E 上的作用有以下 3 种情况：

(1) $\alpha(P)=nP$，其中 $n\in\mathbf{Z}$，此时记 $\alpha-n=0$；

(2) $\alpha^2(P)+a\alpha(P)+b=\infty$，其中 $a,b\in\mathbf{Z}$，此时记 $\alpha^2+a\alpha+b=0$；

(3) α 符合一个 4 次多项式。

例 3.3.2　设 E 是定义在 q 元有限域 $\boldsymbol{F}_q$ 上的椭圆曲线，其上的 q-Frobenius 映射 ϕ_q 定义为

$$\phi_q(x,y)=(x^q,y^q)$$

显而易见，ϕ_q 是 E 上的一个次数为 q 的不可分自同态，且有

$$\phi_q^2-a\phi_q+q=0$$

其中 a 的意义将在下一节中给出详细说明。

例 3.3.3　设素数 $p\equiv 1 \pmod 4$，$i\in\boldsymbol{F}_q$ 是一个乘法 4 阶元素。考虑椭圆曲线 $E/\boldsymbol{F}_q$：$y^2=x^3+ax$ 上的映射

$$\phi:\quad \begin{array}{ccc}(x,y) & \mapsto & (-x,iy)\\ \infty & \mapsto & \infty\end{array}$$

则 ϕ 是 E 的一个自同态，且有

$$\phi^2+1=0$$

例 3.3.4　设素数 $p\equiv 1 \pmod 3$，$\rho\in\boldsymbol{F}_q$ 是一个乘法 3 阶元素。考虑椭圆曲线 $E/\boldsymbol{F}_q$：$y^2=x^3+b$ 上的映射

$$\phi:\quad \begin{array}{ccc}(x,y) & \mapsto & (\rho x,y)\\ \infty & \mapsto & \infty\end{array}$$

则 ϕ 是 E 的一个自同态，且有

$$\phi^2+\phi+1=0$$

本节中的例子都是在椭圆曲线密码学中常常要用到的。

3.4　曲线上点的个数

3.4.1　有限域上椭圆曲线上点的个数

设 E/K 是一条椭圆曲线，令 n 是一个正整数，记

$$E[n]=\{P\in E(\overline{K})\mid nP=\infty\}$$

为 E 上 n 阶点全体。如果 K 的特征不整除 n 或者等于 0，则[1]

$$E[n]\cong\mathbf{Z}_n\oplus\mathbf{Z}_n$$

如果域 K 的特征是素数 p，且 $p\mid n$。记 $n=p^r n'$，$p\nmid n'$，则[1]

$$E[n]\cong\mathbf{Z}_{n'}\oplus\mathbf{Z}_{n'}\quad 或者\quad E[n]\cong\mathbf{Z}_n\oplus\mathbf{Z}_{n'}$$

设 F_q 是 q 元有限域，则定义在 F_q 上的椭圆曲线 E/F_q 上点的个数的最基本结果由以下 Hasse 引理给出。

定理 3.4.1(Hasse 引理)　$|\#E(F_q)-q-1|\leqslant 2\sqrt{q}$。

定理 3.4.1 的证明参见文献[4]。

Hasse 引理表达出这样一个信息：定义在 q 元有限域上的椭圆曲线上点大约有

$q+1$ 个，其确切值和 $q+1$ 的误差的绝对值比 q 小得多，即不超过 $2\sqrt{q}$。

记 $a=q+1-\#E(F_q)$，则 a 是唯一满足

$$\phi_q^2 - a\phi_q + q = 0$$

的整数，因此称 a 为 Frobenius 映射 ϕ_q 的迹，即 $a=\mathrm{Tr}(\phi_q)$。此时多项式 X^2-aX+q 称为 Frobenius 映射 ϕ_q 的特征多项式。

如果 $X^2-aX+q=(X-\alpha)(X-\beta)$，则 $\alpha^n+\beta^n$ 是 ϕ_q^n 的迹，这样就有[5]

$$\#E(F_{q^n}) = q^n + 1 - (\alpha^n + \beta^n)$$

对于比较小的 q，这个结果可以给出一个计算椭圆曲线的阶的方法。

例 3.4.1 定义在 F_2 上的椭圆曲线 $y^2+xy=x^3+1$ 上有 4 个点(0,1)、(1,0)、(1,1)、∞。因此 $a=2+1-4=-1$，于是有

$$X^2 + X + 2 = \left(X - \frac{-1+\sqrt{-7}}{2}\right)\left(X - \frac{-1-\sqrt{-7}}{2}\right)$$

令 $n=101$，则有

$$\left(X - \frac{-1+\sqrt{-7}}{2}\right)^{101} + \left(X - \frac{-1-\sqrt{-7}}{2}\right)^{101} = 2\,969\,292\,210\,605\,269$$

因此

$$\begin{aligned}\#E(\boldsymbol{F}_2^{101}) &= 2^{101} + 1 - 2\,969\,292\,210\,605\,269 \\ &= 2\,535\,301\,200\,456\,455\,833\,701\,195\,805\,484\end{aligned}$$

3.4.2 超奇异椭圆曲线

设 E/K 是一条定义在特征 p 的有限域上的椭圆曲线，如果 $E[p]=\{\infty\}$，则 E 称为是一条超奇异椭圆曲线。换言之，特征 p 的有限域上超奇异椭圆曲线没有 p 阶点。

超奇异椭圆曲线不是奇异曲线。之所以用“奇异”这个词来描述这类椭圆曲线的特点，是因为这类椭圆曲线的自同态环是四元数域的子环，有兴趣的读者请参阅文献[1]。

以下定理可以帮助我们确定一条椭圆曲线是否为超奇异椭圆曲线。

定理 3.4.2 设 $E/\boldsymbol{F}_q$ 是定义在 q 元有限域上的椭圆曲线，其中 q 是素数的方幂。令 $a=q+1-\#E(\boldsymbol{F}_q)$，则 E 是超奇异椭圆曲线当且仅当 $p\mid a$。

推论 设素数 $p\geqslant5$，则 $E/\boldsymbol{F}_q$ 是超奇异椭圆曲线当且仅当 $a=0$，即 $\#E(\boldsymbol{F}_q)=p+1$。

例 3.4.2 以下椭圆曲线均为超奇异椭圆曲线：

(1) $E/\boldsymbol{F}_2: y^2+y=x^3+x$，其中 $\#E(\boldsymbol{F}_2)=5$；

(2) $E/\boldsymbol{F}_3: y^2=x^3-x+2$，其中 $\#E(\boldsymbol{F}_3)=1$；

(3) $E/\boldsymbol{F}_q: y^2=x^3+b$，其中奇数 $q\equiv2 \pmod 3)$，$b\in\boldsymbol{F}_q^{\times}$，$\#E(\boldsymbol{F}_q)=q+1$。

3.4.3 非正常曲线

椭圆曲线 $E/\boldsymbol{F}_q$ 称为非正常曲线，如果 $\#E(\boldsymbol{F}_q)=q$。非正常曲线上点群同构

于 $\boldsymbol{F}_q$ 上的加法群，因此离散对数问题是容易求解的。在选取椭圆曲线构造密码系统时不能使用这类曲线。

3.5　对子

将 Weil 对和 Tate 对统称为对子，也有作者称之为双线性对。

3.5.1　除子

为了定义对子，必须引进一个代数几何的概念：除子。

定义 3.5.1　设 E/K 是定义在 K 上的一条椭圆曲线，E 的 $\overline{K}$ 有理点生成的有限形式和

$$D = \sum_{P\in E(\overline{K})} n_P(P)$$

称为 E 的一个除子，这里 n_P 是整数。

除子 D 的次数定义为 $\deg(D) = \sum_{P\in E(\overline{K})} n_P$，次数为零的除子称为零次除子。

除子 D 的和定义为 $\mathrm{sum}(D) = \sum_{P\in E(\overline{K})} n_P P$，这是椭圆曲线上的一个点。

除子 D 的支集定义为 $\mathrm{supp}(D) = \{P \mid P \in E(\overline{K}), n_P \neq 0\}$，表示除子里包含哪些点。

E 上全体除子所成的自由 Abel 群称为 E 的除子群，记为 $\mathrm{div}(E)$。

在除子的定义中，有限和的意思是对几乎所有的 P 都有 $n_P=0$；而形式和的意思就是只要点在曲线上就可以了，而点的其他数学性质，比如弦切律定义的加法等，都不予考虑。例如，在除子

$$D_1 = (P) + (-P)$$

中，(P)和$(-P)$没有任何关系，它们只是形式地加在一起，因此 $D_1 \neq (\infty)$；而在除子

$$D_2 = 2(P)$$

中，2 也只是形式地乘在(P)上，同样 $D_2 \neq (2P)$。不妨用如图 3.4 所示的示意图来理解除子的概念。

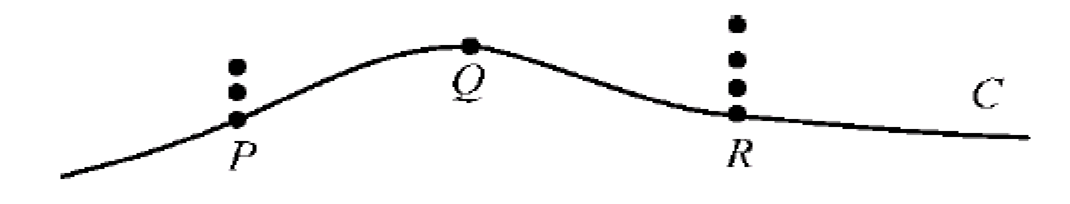


图 3.4　除子的概念示意图

除子：$D=3(P)+(Q)+4(R)$。次数：$\deg D=3+1+4=8$。和：$\mathrm{sum}(D)=3P+Q+4R$。支集：$\mathrm{supp}(D)=\{P,Q,R\}$。

除子的概念比较难以理解，但是除子的构造可以类比于从素数构造有理数。如果只考虑有理数的乘法关系，整数的构造就是

$$n = \prod p_i^{n_i}, \quad n_i \geqslant 0$$

有理数的构造是

$$a = \prod p_i^{n_i}, \quad n_i \in Z$$

除子和有理数的区别有两个：①除子的运算是加法，有理数的运算是乘法；②除子是纯粹的形式和，而有理数的积是有确切值的。

假设 E：$y^2 = x^3 + ax + b$ 是一条椭圆曲线，E 上的函数是指至少在 $E(\overline{K})$ 的一个点上有定义的有理式 $f(x,y) \in \overline{K}(x,y)$。例如，$g(x,y) = \dfrac{1}{y^2 - x^3 - ax - b}$ 就不是 E 上的函数。设有 E 上的函数 $f(x,y) = \dfrac{p(x,y)}{q(x,y)}$，其中 $p(x,y), q(x,y) \in \overline{K}[x,y]$，则作为函数有

$$f(x,y) = \frac{p(x,y)(\bmod\ y^2 - x^3 - ax - b)}{q(x,y)(\bmod\ y^2 - x^3 - ax - b)}$$

E 上的点 P 称为函数 $f(x,y)$ 的零点，如果 $f(P)=0$；点 Q 称为函数 $f(x,y)$ 的极点，如果 $1/f(Q)=0$。类比复变函数中零点和极点的阶的概念，也可以引入 E 上函数的零点和极点的阶的概念。设 P 是 E 上一个点，则存在被称为 P 点的一致化子的函数 u_P，使得 $u_P(P)=0$，且任一函数 $f(x,y)$ 都可以写成以下形式：

$$f = u_P^r g, \quad 其中\ r \in Z，且\ g(P) \neq 0, \infty$$

定义 f 在 P 点的阶为

$$\mathrm{ord}_P(f) = r$$

根据复变函数论，函数的零点和极点只有有限多个。椭圆曲线上的函数实际上就是一类特殊的复变函数，因此也只有有限个零点和极点，故而有以下定义。

定义 3.5.2 设 f 是 E 上一个非零函数，f 定义的除子为

$$\mathrm{div}(f) = \sum_{P \in E(\overline{K})} \mathrm{ord}_P(f)(P) \in \mathrm{Div}(E)$$

函数定义的除子称为主除子。两个除子 D_1、D_2 称为等价的，如果它们之间差一个主除子。等价的除子记为 $D_1 \sim D_2$，即有

$D_1 \sim D_2 \Leftrightarrow$ 存在 E 上函数 f，使得 $D_1 = D_2 + \mathrm{div}(f)$。

根据复变函数的理论，主除子显然是零次除子。但是，零次除子不一定是主除子。

定理 3.5.1 椭圆曲线上的零次除子 $D = \sum_i n_i(P)$ 是主除子当且仅当 $\mathrm{sum}(D) = \infty$。

定理 3.5.1 的证明参见文献[4]。

3.5.2 Weil 对

设 n 和 F_q 的特征 p 互素，E 是定义在 F_q 上的椭圆曲线，且存在正整数 m 符合

$$E[n] \subset E(F_{q^m})$$

以下来定义 Weil 对

$$e_n: E[n] \times E[n] \to \mu_n = \langle \zeta_n \rangle \subset F_{q^m}$$

其中 μ_n 是 n 次单位根群。

设 $P,Q\in E[n]$，令零次除子 $D_1\sim(P)-(\infty)$，$D_2\sim(Q)-(\infty)$ 并且 $\mathrm{supp}(D_1)\cap\mathrm{supp}(D_2)=\varnothing$。由定理 3.4.1 知存在函数 f,g 使得

$$\mathrm{div}(f)=nD_1=n(P)-n(\infty),\quad \mathrm{div}(g)=nD_2=n(Q)-n(\infty)$$

则定义 Weil 对为

$$e_n(P,Q)=\frac{f(D_2)}{g(D_1)}$$

其中函数在除子上的取值定义为：对于除子 $D=\sum_i a_i(Q_i)$，$f(D)=\prod_i f(Q_i)^{a_i}$。

定理 3.5.2　Weil 对具有以下性质。

(1)（双线性）对任意的 $S,S_1,S_2,T,T_1,T_2\in E[n]$，则有

$$e_n(S_1+S_2,T)=e_n(S_1,T)e_n(S_2,T)$$
$$e_n(S,T_1+T_2)=e_n(S,T_1)e_n(S,T_2)$$

(2)（非退化性）如果对任意的 $T\in E[n]$ 都有 $e_n(S,T)=1$，则 $S=\infty$；同样地，如果对任意的 $S\in E[n]$ 都有 $e_n(S,T)=1$，则 $T=\infty$。

(3)（归一性）对任意的 $T\in E[n]$ 都有 $e_n(T,T)=1$。

证明：略，请参见参考文献[4]。

密码学需要的对子应该满足新的非退化性：对 n 阶点 $T\in E[n]$，有 $e_n(T,T)=\zeta_n$。这和 Weil 对的归一性相违背，因此需要对 Weil 对做些改造。

由于 n 和椭圆曲线的定义域 K 的特征互素，由 3.4.1 节可知

$$E[n]\cong\mathbf{Z}_n\oplus\mathbf{Z}_n$$

也就是说，$E[n]=\langle P\rangle\otimes\langle Q\rangle$，其中 P、Q 是 n 阶点。取一个满足条件 σ：$P\mapsto Q$ 的同态映射，如图 3.5 所示，对 Weil 对进行以下改造：

$$\tilde{e}_n(P_1,P_2)=e_n(P_1,\sigma(P_2))$$

则新定义的对子 $\tilde{e}_n$ 符合密码学的要求。以后再说到 Weil 对都是指改造后的 Weil 对。

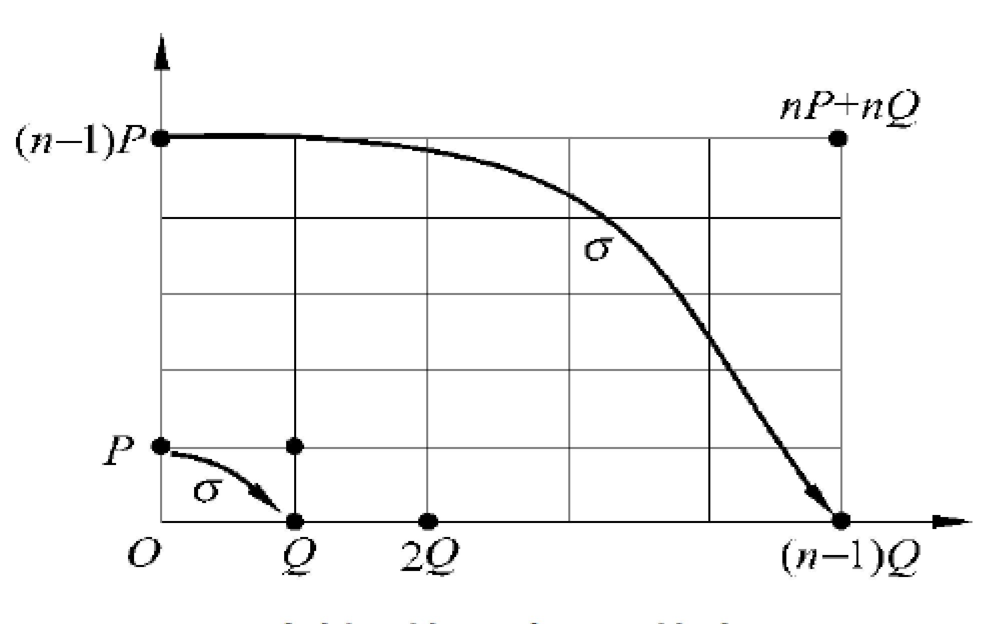


图 3.5　映射σ的示意图（其中 $O=\infty$）

下面的定义说明什么样的同态映射可以用来改造 Weil 对。

定理 3.5.3　设 $P\in E(F_q)$ 是一个 r 阶点，其中 r 是个素数。设整数 $m>1$。如果 $E(F_{q^m})$ 中没有 r^2 阶点。令 σ 是 E 的一个自同态，如果 $\sigma(P)\notin E(F_q)$，则 $e(P,\sigma(P))\neq 1$。

把符合条件 $E[n]\subset E(F_{q^m})$ 的最小正整数 m 称为这条椭圆曲线对于整数 n 的嵌入次数。应该注意到，即使工作在椭圆曲线的 F_q-有理点上，对子的取值还是在扩域 F_{q^m} 中。所以这里嵌入次数的大小对 Weil 对的计算有着重要的意义：嵌入次数越小，Weil 对越容易计算；嵌入次数越大，Weil 对越不容易计算。嵌入次数的值小到一定的程度，比如不大于 2，则用这种椭圆曲线做出来的密码系统容易受到 MOV 攻击；嵌入次数的值大到一定程度，比如大于 30，则计算对子的值就要消耗超量的计算资源，从而一些利用对子的密码系统，比如基于身份的密码系统就没法实现了。超奇

异椭圆曲线的嵌入次数不超过 6[7]，常被用来构造基于对子的密码系统。

以下定理为确定椭圆曲线的嵌入次数提供了一条重要的线索。

定理 3.5.4 设 E/F_q 是一条定义在 F_q 上的椭圆曲线，素数 $\ell \mid \# E(F_q)$，$E[\ell] \not\subset E(F_q)$，且 $\ell \nmid q(q-1)$，则

$$E[\ell] \subset E(F_{q^m}) \quad 当且仅当 \quad q^m \equiv 1 (\bmod\ \ell)$$

这个定理在 Tate 对的理论中起着同样的作用。

3.5.3 Tate 对

设 E/F_q 是一条椭圆曲线，正整数 $n \mid q^m - 1$。可以定义另一种对子，Tate 对：

$$\tau_n: E(F_{q^m})[n] \times E(F_{q^m})/nE(F_{q^m}) \to \mu_n$$

Tate 对也具有双线性、非退化等性质，在密码学中起着和 Weil 对同样的作用。下面就来介绍 Tate 对的构造。

设 $P, Q \in E(F_{q^m})[n]$，则存在函数 f 使得

$$\operatorname{div}(f) = n(P) - n(\infty)$$

再令除子 $D \sim (Q) - (\infty)$ 是一个零次除子，且 $\operatorname{supp}(D) \cap \{P, \infty\} = \varnothing$，则 Tate 对 τ_n 的定义为

$$\tau_n(P, Q) = f(D)^{\frac{q^m - 1}{n}}$$

Tate 对具有双线性和非退化等性质。

定理 3.5.5 以上定义的 Tate 对具有以下性质：

(1)（双线性）对任意的 $S, S_1, S_2 \in E(F_{q^m})[n]$，$T, T_1, T_2 \in E(F_{q^m})/nE(F_{q^m})$，

$$\tau_n(S_1 + S_2, T) = \tau_n(S_1, T)\tau_n(S_2, T)$$

$$\tau_n(S, T_1 + T_2) = \tau_n(S, T_1)\tau_n(S, T_2)$$

(2)（非退化性）如果对任意的 $T \in E(F_{q^m})/nE(F_{q^m})$ 都有 $\tau_n(S, T) = 1$，则 $S = \infty$；同样地，如果对任意的 $S \in E[n]$ 都有 $\tau_n(S, T) = 1$，则 $T = \infty$。

3.5.4 对子的计算

以 Tate 对的计算为例，其关键是计算函数 f，使得 $\operatorname{div}(f) = n(P) - n(\infty)$。Miller 算法给出了解决这个问题的基础。Miller 算法的目的是计算函数 f_i，使得

$$\operatorname{div}(f_i) = i(P) - (iP) - (i-1)(\infty)$$

这样 $f = f_n$。

注意到对于两个椭圆曲线上的函数 f, g，有 $\operatorname{div}(fg) = \operatorname{div}(f) + \operatorname{div}(g)$。利用这个关系构造计算 f_i 的一个递归算法。

在计算点 $2P$ 的过程中，记 l_P 是过 P 点的切线，v_P 是过 $-2P$ 和 $2P$ 的直线。把 l_P 和 v_P 看成椭圆曲线上的函数，则

$$f_2 = l_P / v_P \tag{3.7}$$

即

$$\operatorname{div}\left(\frac{l_P}{v_P}\right) = 2(P) - (2P) - (\infty)$$

假设 f_i 和 f_j 已经得到，记 $l_{i,j}$ 是过 iP 点和 jP 点的直线，$v_{i,j}$ 是过 $-(i+j)P$ 点和 $(i+j)P$ 点的直线。把 $l_{i,j}$ 和 $v_{i,j}$ 看成椭圆曲线上的函数，则

$$f_{i+j} = f_i f_j l_{i,j} / v_{i,j} \tag{3.8}$$

即

$$\mathrm{div}\left(f_i f_j \frac{l_{i,j}}{v_{i,j}}\right) = (i+j)(P) - ((i+j)P) - (i+j-1)(\infty)$$

图 3.6 和图 3.7 分别是计算 f_2 和 f_{i+j} 的示意图。下面以图中的直线 l_P 为例来说明算法中函数的意义。

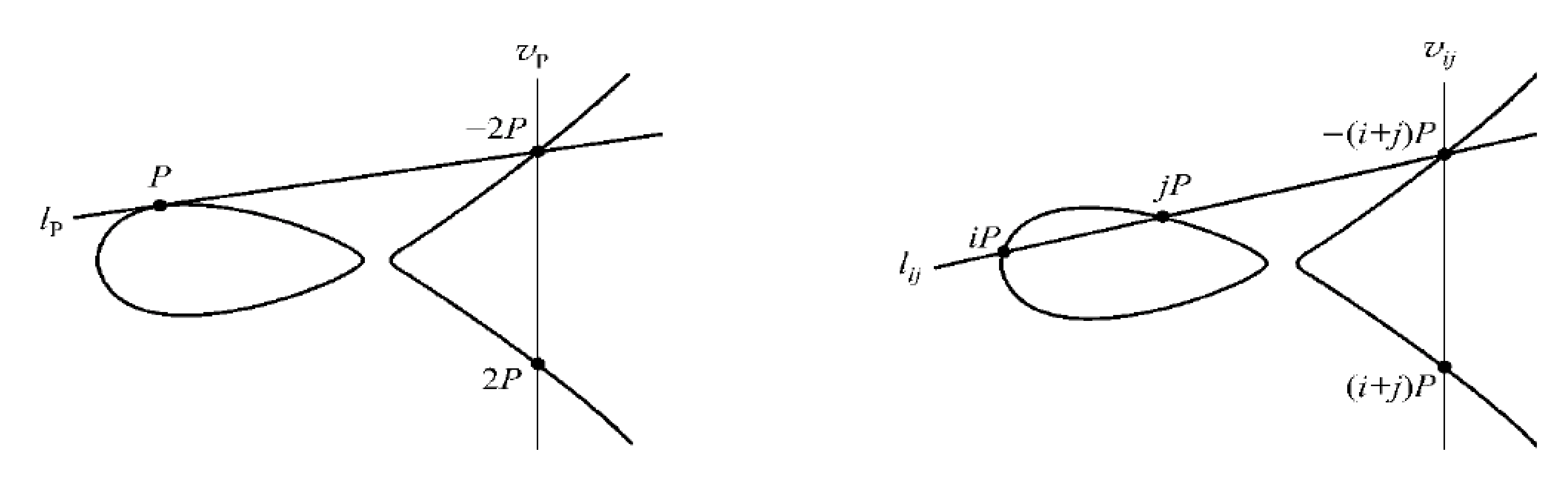


图 3.6　计算 f_2 的示意图　　　图 3.7　计算 f_{i+j} 的示意图

l_P 与椭圆曲线相交于 P 点两次，$-2P$ 点 1 次，而且没有其他交点，因此 P 是 l_P 的 2 阶零点，$-2P$ 是 l_P 的 1 阶零点且没有其他零点。显然 l_P 没有有限的极点，因此其极点为无穷远点，又根据函数的极点和零点个数相等可知，无穷远点是 l_P 的 3 阶极点，故而有

$$\mathrm{div}(l_P) = 2(P) + (-2P) - 3(\infty)$$

同样的分析可知

$$\mathrm{div}(v_P) = (-2P) + (2P) - 2(\infty)$$

因此

$$\begin{aligned}\mathrm{div}(l_P/v_P) &= [2(P) + (-2P) - 3(\infty)] - [(-2P) + (2P) - 2(\infty)] \\ &= 2(P) - (2P) - (\infty)\end{aligned}$$

把 n 用二进制表示，利用倍点相加的技术，反复利用式(3.7)和式(3.8)可以在 $\lg n$ 的多项式时间内得到 f。Weil 对的计算类似，请读者参考文献[6]。Miller 算法保证了对子的可计算性，也就保证了基于椭圆曲线上对子的密码学体制在应用上的可行性。

3.6　椭圆曲线密码体制

本文涉及的椭圆曲线密码体制是基于下列困难问题：

(1) Diffie-Hellman(DH)问题。设 E 是一条椭圆曲线，P 是 E 上一个 r 阶点，其中 r 是一个大素数。则称 E 上 DH 问题是一个困难问题，如果对任意的多项式时间

算法 A，对于给定 E 上点 aP、bP，以下概率可忽略：

$$\mathrm{Adv}(A) = \Pr[abP = A(P, aP, bP)]$$

（2）双线性 Diffie-Hellman（BDH）问题。设 E 是一条椭圆曲线，P 是 E 上一个 r 阶点，其中 r 是一个大素数，e 是 E 上定义的可以有效计算的一个对子。则称 E 上 BDH 问题是一个困难问题，如果对任意的多项式时间算法 A，对于给定 E 上点 aP、bP、cP，以下概率可忽略：

$$\mathrm{Adv}(A) = \Pr[e(P, P)^{abc} = A(P, aP, bP, cP)]$$

对于不同的应用环境，椭圆曲线上可以定义许多种不同的困难问题，这里只列出两个最基本的。下面以上述两个困难问题为基础构造一些密码学协议。以下几节中的 E、P、r 的含义与本节相同。

3.6.1 Diffie-Hellman（DH）密钥交换协议

如果 Alice（A）和 Bob（B）希望利用公开信道建立一个公共的会话密钥，则他们可以利用以下协议：首先共同选择一条椭圆曲线，这条椭圆曲线上的 DH 问题是个困难问题。然后分别选取随机数 a、b，Alice 计算 aP 并发给 Bob；Bob 计算 bP 并发给 Alice。Alice 收到 bP 后计算 $a(bP)$ 得到密钥 abP，Bob 收到 aP 后计算 $b(aP)$ 也可以得到 abP，于是他们之间建立起一个公共的密钥。图 3.8 所示是 DH 密钥交换协议的示意图。

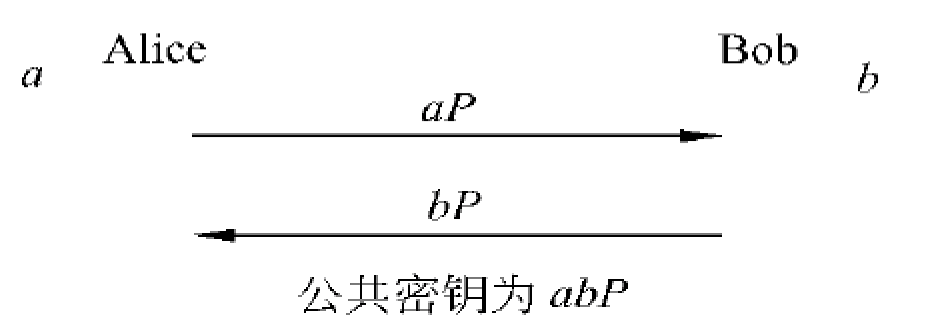


图 3.8　DH 密钥交换协议示意图

利用对子还可以用一轮信息交换建立三方会话密钥。

Alice（A）、Bob（B）和 Charlie（C）希望通过公开信道建立一个三方的公共会话密钥。他们首先共同选择一条椭圆曲线，这条椭圆曲线上的 BDH 问题是个困难问题。然后分别选取随机数 a、b、c，Alice 计算 aP 分别发给 Bob 和 Charlie；Bob 计算 bP 分别发给 Alice 和 Charlie；Charlie 计算 cP 分别发给 Alice 和 Bob。Alice 收到 bP、cP 后计算 $e(bP, cP)^a$ 得到密钥 $e(P, P)^{abc}$，Bob 和 Charlie 类似也可以得到 $e(P, P)^{abc}$，于是他们之间建立起一个公共的密钥。图 3.9 所示是三方密钥交换协议的示意图。

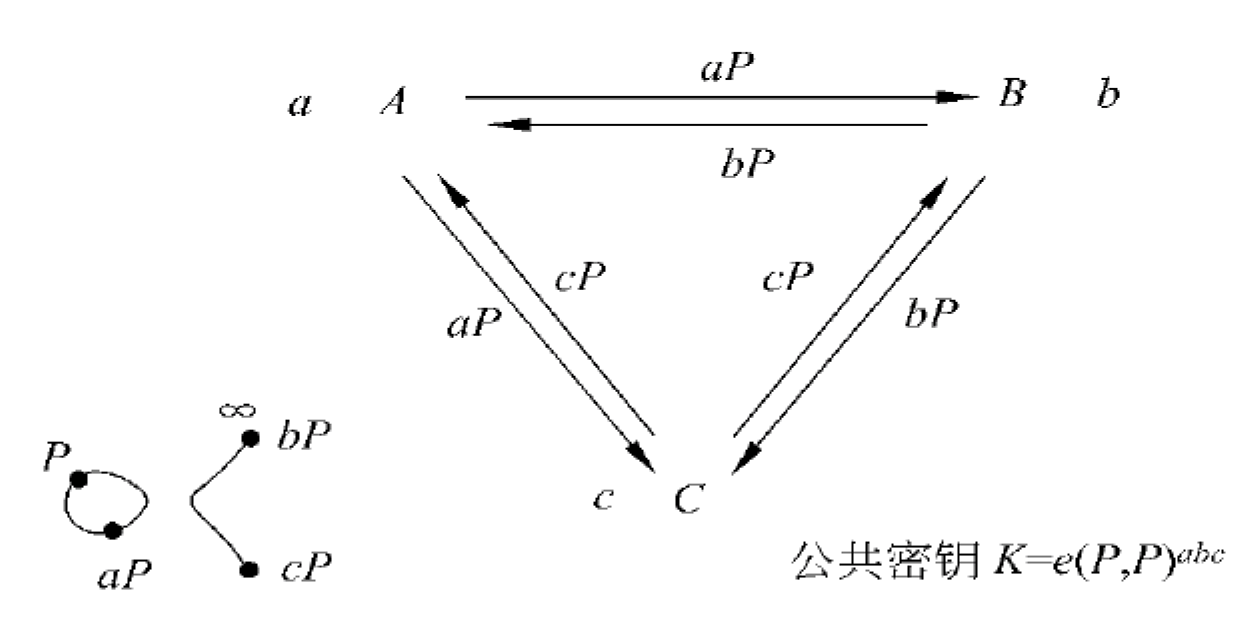


图 3.9　一轮三方 D-H 密钥交换协议

3.6.2　基于身份的密码体制

与 DH 密钥交换协议一样,只要不涉及对子的应用,椭圆曲线上的密码体制基本上和基于有限域上离散对数问题的密码体制类似,这里就不再逐一详细描述。本节主要介绍最近比较受到关注的基于身份的密码体制。

Alice 要给 Bob 发电子邮件,她希望利用 Bob 的公钥对邮件进行加密。首先她需要确定在网上找到的 Bob 的公钥确实是 Bob 公布的,而不是别人假冒的。这就需要 Bob 的密钥与 Bob 的身份进行绑定。解决密钥和身份绑定的方案是由 CA 给 Bob 颁发证书。利用对子,可以构造一种身份和密钥自然绑定的密码体制,也就是基于身份的公钥密码体制。

首先需要一个密钥分发中心 S,这个中心选取一条安全的椭圆曲线,在这条椭圆曲线上 BDH 问题是困难的。S 选取一个随机数 s 作为整个系统的最高机密加以保护,并公布 sP。如果 Alice 要给 Bob 发送明文 M,则她从公开信息中获取 sP 和 Bob 的身份 $ID_B \in E$。加密时 Alice 选取随机数 t 并计算 $U = tP, V = M \oplus e(ID_B, sP)^t$,把 (U, V) 作为密文发送给 Bob。

Bob 接到密文后先向 S 索取自己的私钥,S 把 sID_B 发给 Bob 作为私钥。Bob 拿到私钥后计算 $M' = V \oplus e(U, sID_B)$,则 M' 即为解密后的明文,如图 3.10 所示。

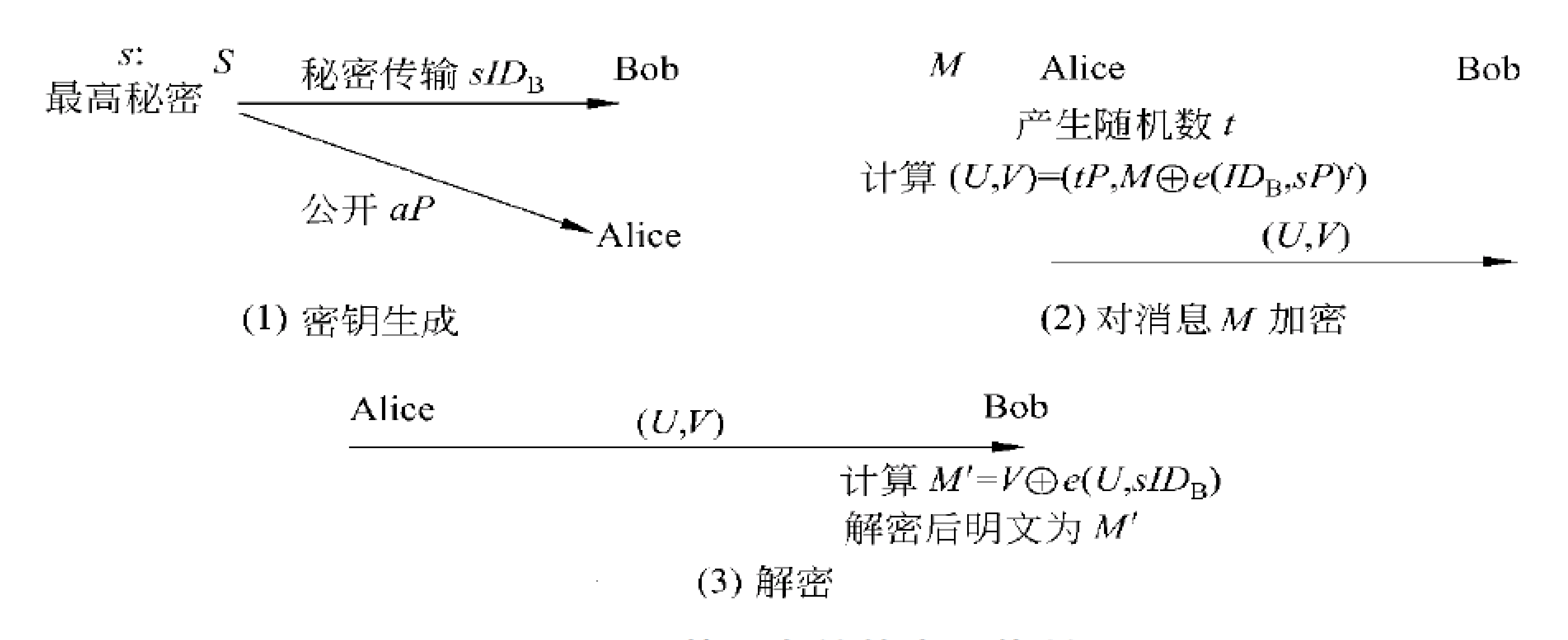


图 3.10　基于身份的密码体制

这是最基本的基于身份的方案。更多基于身份的密码学,请参见文献[7]。

3.7　点标量乘法的计算

点的倍乘 nP,也就是点 P 自己加 n 次,是椭圆曲线上的基本运算。点的倍乘没有统一的术语,有的文献叫点乘;也有的文献叫点的标量乘法,之所以叫标量乘法是把点看成向量,而把点群看成是整数环上的向量空间(也就是整数环上的代数)。这里用点乘这个术语。

点乘是椭圆曲线密码体制实施过程中最常见、最耗时间的运算,一般涉及的 n 都是比较大的整数。因此,实现高效、快速的点乘算法是椭圆曲线密码体制实现的一个

关键。要提高点乘算法的速度，一般来说有3个角度需要考虑：提高有限域上算术运算的速度；提高一次点加运算的速度；减少点加次数。

用$|n|$来表示n的二进制展开的长度，则把n表示成二进制展开再利用倍点相乘的算法就可以得到一个时间复杂度为$O(|n|)$的算法。

在椭圆曲线中$-(x,y)=(x,-y)$，所以最基本的加速方法就是把倍点相加算法用这个关系式进行优化，给出n的非连接形式(NAF)表示：

$$n = a_0 + a_1 2 + a_2 2^2 + \cdots + a_s 2^s \quad a_i \in \{-1,0,1\}, a_i a_{i+1} = 0$$

用n的非连接形式(NAF)的点加运算可以比倍点相加算法快1/3。

点加的快速算法是椭圆曲线密码学的一个基本问题，有大量的文献研究，这里就不一一列出。

3.8 注记

本章主要围绕椭圆曲线在信息安全中的应用，介绍了椭圆曲线的一些基本概念和基本原理。有关椭圆曲线的更多理论与结果，可参考文献[3]、[4]，有关椭圆曲线在密码学中的应用可参考文献[1]、[2]和[6]。近年来，椭圆曲线在整数的素性检测和因子分解方面也有许多成功的应用，有兴趣的读者可参看文献[5]。关于椭圆曲线的理论及其在密码学中的应用还在发展，近年来越来越多的人研究超椭圆曲线在密码学中的应用，有关超椭圆曲线在密码学中的应用，读者可参看文献[7]。

参考文献

[1] Blake I F, Seroussi G, Smart N P. Elliptic Curves in Cryptography. Cambridge University Press, 1999

[2] Blake I F, Seroussi G, Smart N P. Advances in Elliptic Curve Cryptography. Cambridge University Press, 2005

[3] Knapp A W. Elliptic curves, volume 40 of Mathematical Notes, Princeton University Press, Princeton, NJ, 1992

[4] Silverman J H. The Arithmetic of Elliptic Curves, Springer-Verlag, New York,1986

[5] Washington L C. Elliptic Curves Number Theory and Cryptography, Chapman & Hall/CRC, 2003

[6] 王学理，裴定一. 椭圆与超椭圆曲线公钥密码的理论与实现. 北京：科学出版社，2006

[7] Heri Cohen, Gerhard Frey. Handbook of Elliptic and Hyperelliptic Curve Cryptography. Discrete Mathematics and its application. Chapman & Hall/CRC, 2006

第4章　组合论方法与技术

组合数学是当代数学中非常重要的一个分支，它的历史渊源扎根于数学娱乐和游戏之中。过去不论是出于消遣还是出于其美学上的魅力而被研究的许多组合问题，在当今的纯粹科学和应用科学研究中都具有很重要的价值。随着计算机科学的飞速发展，组合数学有了更强的生命力。由于运算速度的不断增加，计算机已经能够解决以前不敢想象的大型问题。然而计算机的运行需要通过程序来控制，这些程序的基础往往是求解问题的组合数学算法，对于这些算法，运行时间效率和存储需求分析需要更多的组合数学思想。在现代，组合数学的思想和技巧不仅在计算机科学领域，而且在实验设计、工业规划、电子通信和信息安全等诸多重要领域都有着广泛而重要的应用。

组合数学是研究离散结构的存在、计数、分析和优化等问题的一门学科。组合数学的基本问题是：存在问题、计数问题和优化问题。存在问题研究的是是否存在一种特定种类的排列；计数问题研究的是存在多少种排列；优化问题所关心的是在各种可能的排列中，选择对于某个标准来说最好的排列。

本章主要介绍组合数学中的一些基本概念和基本原理，以及拉丁方与区组设计的一些基本性质，最后给出了在信息安全中的几个应用实例。

4.1　基本计数原理、排列与组合

4.1.1　基本计数原理

组合数学的基础是由若干基本计数原理组成。第一个原理是非常基本的，它是整体等于其部分之和这一原理的公式化。

设 S 是一个集合，$S_1,S_2,\cdots,S_m$ 是 S 的子集合。如果 $S_1,S_2,\cdots,S_m$ 满足：

$$S=S_1\cup S_2\cup\cdots\cup S_m,\quad S_i\cap S_j=\phi(i\neq j)$$

则称 $\{S_1,S_2,\cdots,S_m\}$ 构成集合 S 的一个**划分**，子集 $S_1,S_2,\cdots,S_m$ 称为该划分的**部分**。集合 S 的元素个数表示为 $|S|$，也称为 S 的**大小**。

加法原理　设 $\{S_1,S_2,\cdots,S_m\}$ 是集合 S 的一个划分，则要确定 S 中事物的个数，可先求出各部分 $S_1,S_2,\cdots,S_m$ 中事物的个数然后加起来，即

$$|S|=|S_1|+|S_2|+\cdots+|S_m|$$

这里需要说明的是，如果集合 $S_1,S_2,\cdots,S_m$ 可以重叠，要计算 S 中事物的个数则需要更深刻的原理——容斥原理，这个原理将在 4.2 节中讨论。

加法原理的另一种表述　若有 p 种方法从一堆事物中选取一物，而有 q 种方法从另一堆事物中选取一物，则从两堆中选取一物的方法共有 $p+q$ 种。这种方法可以推广到两堆以上的情形。

例 4.1.1 已知从甲地到乙地每天有 3 班船，5 趟汽车，2 趟火车和 4 班飞机。问从甲地到乙地每天有多少种出行方式。

解：利用加法原理知，从甲地到乙地的出行方式共有：3+5+2+4=14 种。

乘法原理 如果 A 是 p 个事物的集合，B 是 q 个事物的集合，$a\in A$，$b\in B$，则形如 (a,b) 的有序对的个数等于 $p\times q$。

乘法原理的另一种表述 如果第一种事物有 p 种选择方式，并且不论第一个事物怎样选择，第二个事物都有 q 种选择方式，那么同时选择第一个事物和第二个事物的方式共有 $p\times q$ 种。乘法原理可以推广到任意有限个集合上。

例 4.1.2 由数字{1,2,3,4,5,6,7,8,9}可以构造多少个 5 位数，使所有数字或者都是奇数或者都是偶数？

解：因{1,2,3,4,5,6,7,8,9}中有 4 个偶数，5 个奇数，则由乘法原理，得知

所有数字都是奇数的 5 位数的个数是：

$$5\times5\times5\times5\times5=3125$$

所有数字都是偶数的 5 位数的个数是：

$$4\times4\times4\times4\times4=1024$$

于是由加法原理，所有数字或者都是奇数或者都是偶数的 5 位数的个数是：

$$3125+1024=4149$$

4.1.2 集合的排列

在组合数学中，经常会碰到由 n 个不同元素组成的集合，为了方便，称它们为 ***n* 集合**。

给定一个 n 集合，假设要从中挑出 r 个元素，并依次排列它们，这样的一个排列称为这个 n 集合的一个 ***r* 排列**。用 $P(n,r)$ 表示 n 集合的 r 排列的个数。如果 $n=r$，则 n 集合的一个 n 排列简称为 n 个元素的一个排列。如果 $n<r$，则 $P(n,r)=0$，在这种情况下，不存在 n 集合的 r 排列。因此在以下的讨论中，如不特殊说明，都假设 $n\geqslant r$。

定理 4.1.1 n 集合的 r 排列的个数为：$P(n,r)=n\times(n-1)\times\cdots\times(n-r+1)$。

证明：在构造 n 集合的一个 r 排列时，可以有 n 种方法选择第一项；只要选出了第一项，就有 $n-1$ 种方法选出第二项；……；而只要选出了前 $r-1$ 项，就有 $n-r+1$ 种方法选出第 r 项。根据乘法原理，这 r 项可以有 $n\times(n-1)\times\cdots\times(n-r+1)$ 种方法选出，即

$$P(n,r)=n\times(n-1)\times\cdots\times(n-r+1)$$

定义 $n!$（读作 n 的阶乘）为

$$n!=n\times(n-1)\times\cdots\times2\times1$$

并规定 $0!=1$。于是有

$$P(n,r)=\frac{n!}{(n-r)!}$$

对于 $n\geqslant0$，定义 $P(n,0)=1$，这与上式中 $r=0$ 时的公式一致。n 个元素的排列

个数为

$$P(n,n)=n!$$

例 4.1.3　现有 20 人参加面试，问有多少种不同的顺序安排前面 5 个人的面试？

解：显然这是一个排列问题，安排前面 5 个人的面试共有下面这么多种顺序：

$$P(20,5)=20\times19\times18\times17\times16=1\ 860\ 480$$

例 4.1.4　一个校园的电话分机有 4 位数字，如果不重复使用数字，可以有多少部分机：

(1) 第一个数字不能是 0？

(2) 第一个数字不能是 0 且第二个数字不能是 1？

解：如果不考虑(1)、(2)的限制条件，不重复使用数字的分机数为：

$$P(10,4)=10\times9\times8\times7=5040$$

(1) 第一个数字是 0 的分机可以有 $P(9,3)=9\times8\times7=504$ 部(因为不能重复使用数字)，所以第一个数字不是 0 的分机部数为：

$$P(10,4)-P(9,3)=10\times9\times8\times7-9\times8\times7=4536$$

(2) 同理，第一个数字不是 0 且第二个数字不是 1 的分机部数为：

$$P(10,4)-P(8,2)=10\times9\times8\times7-8\times7=4984$$

确切地说，刚才讨论的排列应该叫做**线排列**。因为是把元素排成一条线。如果不把元素排成一条线，而排成一个圆，那么排列的个数将会减少。例如，123456 和 234561 是两个不同的线排列，但如果把它们的首尾数字向上连接，则形成同一个圆排列

```
    1
 2     6
 3     5
    4
```

从这个例子可以看出，对上述这个圆排列，从不同的地方断开，可以形成 6 个不同的线排列，所以 1、2、3、4、5、6 这 6 个数形成的圆排列的个数为$\frac{6!}{6}=5!$。一般地，有下面的定理。

定理 4.1.2　n 集合的 r 圆排列的个数为：$\frac{P(n,r)}{r}=\frac{n!}{r(n-r)!}$。特别地，$n$ 个元素的圆排列的个数为$(n-1)!$。

证明：证明本质上已经包含在上一段的论述中。设 S 是一个 n 集合，S 的每一个 r 圆排列可以通过下面的方式得到：

(1) 从 S 中选出 r 个元素做线排列，共有 $P(n,r)$种排列；

(2) 把 $P(n,r)$种线排列按下面方法划分成部分：两个 r 线排列在同一个部分中当且仅当这两个线排列的首尾数字向上连接形成同一个圆排列。则 r 圆排列的个数就等于这种划分的部分的个数。

从前一段的讨论可知，每一部分都含有 r 个 r 线排列，因此，部分的个数就是

$$\frac{P(n,r)}{r}=\frac{n!}{r(n-r)!}$$

例 4.1.5 6 位先生和 6 位女士围桌就座，如果要求男、女交替安排座位，试问有多少种可能的坐法？

解：因为男女交替就座，可让 6 位先生先就座，然后 6 位女士分别插在他们之间。6 位先生就座是圆排列问题，其排列数为：$(6-1)!=5!$，6 位女士插入（注意此时不是圆排列而是线排列问题）共有 $6!$ 种方法，根据乘法原理，就座方法数为

$$5!\times 6!=86400$$

4.1.3 集合的组合

n 集合的一个 **r 组合**就是从这个集合中不考虑次序地取出 r 个元素的一种取法。因此 n 集合的一个 r 组合就是这个集合的一个 r 元子集。用 $C(n,r)$ 表示 n 集合的 r 组合的个数。为方便起见，规定 $C(0,0)=1$。对于正整数 n，易知下列事实是正确的，$C(n,0)=1$，$C(n,1)=n$，$C(n,n)=1$。另外，如果 $n=0$ 且 r 是正整数，则 $C(0,r)=0$。特别要说明的是，如果 $r>n$，则 $C(n,r)=0$，在这种情况下，不存在 n 集合的 r 组合。因此在以下的讨论中，如不特殊说明，都假设 $n\geqslant r$。

定理 4.1.3 对于 $0\leqslant r\leqslant n$，$P(n,r)=C(n,r)\times P(r,r)$，因此 $C(n,r)=\frac{n!}{r!(n-r)!}$。

证明：设 S 是一个 n 集合，S 的每一个 r 排列可以通过下面的方式得到：

(1) 从 S 中选出 r 个元素，共有 $C(n,r)$ 种选择方法；

(2) 将所选出的 r 个元素以某种顺序排列，共有 $P(r,r)$ 种排列方法。

由乘法原理，有 $P(n,r)=C(n,r)\times P(r,r)$，再利用公式 $P(n,r)=\frac{n!}{(n-r)!}$，得到

$$C(n,r)=\frac{P(n,r)}{P(r,r)}=\frac{n!}{r!(n-r)!}$$

例 4.1.6 设空间有 25 个点，其中任意 4 个点都不共面。问它们能确定多少个三角形？多少个四面体？

解：由于没有 4 个点共面，所以不可能有 3 个点共线，因而任何 3 个点都可以组成一个三角形，任何 4 个点都可以组成一个四面体。于是 25 个这样的点所构成的三角形的个数为：

$$C(25,3)=\frac{25!}{3!(25-3)!}=2300$$

25 个这样的点所构成的四面体的个数为：

$$C(25,4)=\frac{25!}{4!(25-4)!}=12650$$

定理 4.1.4 对于 $0\leqslant r\leqslant n$，$C(n,r)=C(n,n-r)$。

证明：$C(n,r)=\frac{n!}{r!(n-r)!}=\frac{n!}{(n-r)![n-(n-r)]!}=C(n,n-r)$。

说明：数$\dfrac{n!}{r!(n-r)!}$也可记为$\dbinom{n}{r}$，称为二项式系数，这是因为这个数出现在二项式展开式中(参见定理 4.1.9)。数$\dbinom{n}{r}$有许多重要的性质，将在 4.1.6 小节中进行讨论。

4.1.4　重集的排列

重集类似于集合，只是它的成员不必是不同的。例如，重集 $M=\{a,a,a,a,b,b,c,c,c,d,d\}$有 11 个元素：4 个 a，2 个 b，3 个 c，2 个 d。也可采用指明不同元素出现的次数来表示一个重集，于是 M 也可用$\{4\cdot a,2\cdot b,3\cdot c,2\cdot d\}$来表示，其中的 4、2、3、2 分别是重集 M 中元素的重复数，一般情况下，通过上下文就足以断定表达式中的圆点是表示重集元素的重复数还是相乘。集合就是所有重复数都等于 1 的重集。当没有关于事物重复数的限制时，允许重集中的事物出现无限多次。例如，a 和 c 出现无限多次，而 b 和 d 分别出现 4 次和 7 次的重集表示为$\{\infty\cdot a,4\cdot b,\infty\cdot c,7\cdot d\}$。

设 S 是重集，则 S 的一个 r 排列就是 S 中 r 个元素的一个有序排列。

定理 4.1.5　设 S 是包含 k 个不同元素而每个元素都具有无限重复数的重集，则S 的 r 排列的个数是 k^r。

证明：在构造 S 的 r 排列时，第一项有 k 种选法，第二项有 k 种选法，……，第 r 项也有 k 种选法，而每一项的选择不依赖于前一项的选择。由乘法原理，r 个项有 k^r 种选法。

例 4.1.7　有 3 种不同颜色的球，假设每种颜色的球都有足够多。现在把这些球放入 4 个不同的盒子里，要求每个盒子放入且只能放入一个球。问有多少种放入的方法?

解：把这 3 种不同颜色的球分别用 1、2、3 来表示，则这个问题可以看成是求重集$\{\infty\cdot 1,\infty\cdot 2,\infty\cdot 3\}$的 4 排列的个数。由定理 4.1.5，这个数等于：$3^4=81$。

现在来计算每个元素具有有限重复数的重集的排列。

定理 4.1.6　设 S 是具有有限重复数 $n_1,n_2,\cdots,n_k$ 的重集且 $n=n_1+n_2+\cdots+n_k$，则 S 的元素的排列个数等于$\dfrac{n!}{n_1!n_2!\cdots n_k!}$。

证明：设重集 S 有 k 个不同的元素 $a_1,a_2,\cdots,a_k$，其重复数分别为 $n_1,n_2,\cdots,n_k$，因此 S 共有 $n=n_1+n_2+\cdots+n_k$ 个元素。需要确定这 n 个元素的排列个数。假设有 n 个位置，现在要把 S 中的 n 个元素放入这 n 个位置。先放 a_1，因 S 中有 n_1 个 a_1，所以要从这 n 个位置中选出 n_1 个位置放 a_1，共有 $C(n,n_1)$种选法。其次放 a_2，要从剩下的 $n-n_1$ 个位置中选出 n_2 个位置放 a_2，共有 $C(n-n_1,n_2)$种选法。接着从它的 $n-n_1-n_2$ 个位置中选出 n_3 个位置放 a_3，共有 $C(n-n_1-n_2,n_3)$种选法。如此做下去并利用乘法原理，得到 S 的排列个数为

$$C(n,n_1)C(n-n_1,n_2)C(n-n_1-n_2,n_3)\cdots C(n-n_1-n_2-\cdots-n_{k-1},n_k)$$
$$=\frac{n!}{n_1!(n-n_1)!}\cdot\frac{(n-n_1)!}{n_2!(n-n_1-n_2)!}$$

$$\cdot \frac{(n-n_1-n_2)!}{n_3!(n-n_1-n_2-n_3)!} \cdots \frac{(n-n_1-\cdots-n_{k-1})!}{n_k!(n-n_1-n_2-\cdots-n_k)!}$$

$$= \frac{n!}{n_1!n_2!\cdots n_k!}$$

例 4.1.8 节日期间某大楼上挂着 15 面彩旗(排成一行),其中红、黄、蓝、绿、紫各 3 面。问这些彩旗共有多少种排列方式?若不允许有两面蓝旗相邻,问有多少种排列方法?

解:15 面彩旗可以看成重集 $S=\{3\cdot a,3\cdot b,3\cdot c,3\cdot d,3\cdot e\}$,由定理 4.1.6 知其排列方法数为

$$\frac{15!}{3!3!3!3!3!} = \frac{15!}{6^5}$$

若有两面蓝旗相邻,则可以把这两面旗子看作一面,而一面蓝旗又可以与其余两面蓝旗中的任何一面相邻,于是其排列数为

$$2 \times \frac{14!}{3!3!2!3!3!} = \frac{14!}{6^4}$$

但在上面的计算中,把三面蓝旗相邻的情况多计算了一次,而三面蓝旗相邻的排列数为

$$\frac{13!}{3!3!1!3!3!} = \frac{13!}{6^4}$$

因此,没有两面蓝旗相邻的排列数为

$$\frac{15!}{6^5} - \frac{14!}{6^4} + \frac{13!}{6^4} = \frac{22 \times 13!}{6^4}$$

若 $r<n$,则一般情况下不存在计算 S 的 r 排列个数的简单公式。但通过使用生成函数的技巧可以得到问题的解,有兴趣的读者可参考文献[1]。在某些情况下可以像下面的例子那样讨论。

例 4.1.9 设 $S=\{3\cdot a,2\cdot b,4\cdot c\}$是含有 9 个元素的重集,求 S 的 8 排列个数。

解:S 的 8 排列可分成 3 个部分求:

(1) $S_1=S\backslash\{a\}=\{2\cdot a,2\cdot b,4\cdot c\}$的 8 排列个数$\frac{8!}{2!2!4!}=420$;

(2) $S_2=S\backslash\{b\}=\{3\cdot a,1\cdot b,4\cdot c\}$的 8 排列个数$\frac{8!}{3!1!4!}=280$;

(3) $S_3=S\backslash\{c\}=\{3\cdot a,2\cdot b,3\cdot c\}$的 8 排列个数$\frac{8!}{3!2!3!}=560$。

因此 S 的 8 排列个数为:$420+280+560=1260$。

4.1.5 重集的组合

设 S 是重集,则 S 的一个 r 组合就是从 S 中不计次序地选取 r 个元素。于是 S 的一个 r 组合本身也是一个重集(S 的子重集)。

定理 4.1.7 设 S 是包含 k 个不同元素而每个元素具有无限重复数的重集,则

S 的 r 组合的个数等于 $C(k-1+r,r)$。

证明：设重集 $S=\{\infty\cdot a_1,\infty\cdot a_2,\cdots,\infty\cdot a_k\}$，其中 $a_1,a_2,\cdots,a_k$ 互不相同。则 S 的任一个 r 组合形如 $\{x_1\cdot a_1,x_2\cdot a_2,\cdots,x_k\cdot a_k\}$，其中 $x_1,x_2,\cdots,x_k$ 为非负整数，且 $x_1+x_2+\cdots+x_k=r$。反之，满足 $x_1+x_2+\cdots+x_k=r$ 的非负整数的每个序列 $x_1,x_2,\cdots,x_k$ 对应 S 的一个 r 组合。因此，r 组合的个数等于方程 $x_1+x_2+\cdots+x_k=r$ 的非负整数解的个数。下面证明，这些非负整数解的个数等于重集 $T=\{r\cdot 1,(k-1)\cdot 0\}$ 的排列的个数。

给定 T 的一个排列，这 $k-1$ 个 0 把 r 个 1 分成 k 组。设有 x_1 个 1 在第一个 0 的左边，x_2 个 1 在第一个 0 和第二个 0 之间，……，x_k 个 1 在最后一个 0 的右边，则 $x_1,x_2,\cdots,x_k$ 为非负整数且 $x_1+x_2+\cdots+x_k=r$。反之，给定非负整数 $x_1,x_2,\cdots,x_k$ 且满足 $x_1+x_2+\cdots+x_k=r$，按与上面相反的步骤可以构造出 T 的一个排列。于是重集 S 的 r 组合的个数等于重集 $T=\{r\cdot 1,(k-1)\cdot 0\}$ 的排列的个数。由定理 4.1.6 知，T 的排列个数等于

$$\frac{(k-1+r)!}{(k-1)!r!}=C(k-1+r,r)$$

例 4.1.10　一家面包房生产 6 种不同的面包，如果要买一打(12 只)面包，可有多少种不同的选择方案？如果还要求每种面包至少选择一个，有多少种选择方案？这里假定面包房每种面包的数量远远大于 12 只。

解：面包房的面包可以表示为：$S=\{\infty\cdot a_1,\infty\cdot a_2,\cdots,\infty\cdot a_6\}$，每一打面包相当于 S 的一个 12 组合，由定理 4.1.7 知，买一打面包的选择方案数为

$$C(6-1+12,12)=6188$$

由定理 4.1.7 的证明知，每种面包至少选择一个的选择数等于方程

$$x_1+x_2+\cdots+x_6=12$$

的正整数解的个数。作变量替换

$$y_1=x_1-1,y_2=x_2-1,\cdots,y_6=x_6-1$$

则上面的方程变为

$$y_1+y_2+\cdots+y_6=6$$

这里 $y_1,y_2,\cdots,y_6$ 是非负整数。由定理 4.1.7 知，新方程的非负整数解的个数为

$$C(6-1+6,6)=462$$

所以买一打面包且每种面包至少选择一个的选择方案为 462 种。

4.1.6　二项式展开

在 4.1.3 小节中曾对所有非负整数 k 和 n 定义了二项式系数 $\binom{n}{k}$，并且已知下面的结果：$\binom{n}{0}=1$，$\binom{n}{1}=n$，$\binom{n}{n}=1$；若 $n=0$ 且 k 为正整数，则 $\binom{0}{k}=0$；若 $k>n$，则 $\binom{n}{k}=0$；对 $0\leqslant k\leqslant n$，$\binom{n}{k}=\frac{n!}{k!(n-k)!}$，且 $\binom{n}{k}=\binom{n}{n-k}$。

定理 4.1.8(Pascal 公式) 对于满足 $1\leqslant k\leqslant n-1$ 的整数 k 和 n,有

$$\binom{n}{k}=\binom{n-1}{k}+\binom{n-1}{k-1}$$

证明:证明这个恒等式的一种方法是直接代入公式 $\binom{n}{k}=\frac{n!}{k!(n-k)!}$,验证等式两边是否相等即可。

下面给出一个组合证法。令 S 是一个 n 集合,a 是 S 中的一个元素。现在利用 a 把 S 的 k 组合集 T 划分成两部分 A 和 B,其中 A 包含 S 的所有不含 a 的 k 组合,B 包含 S 的所有含 a 的 k 组合。则由加法原理,$|T|=|A|+|B|$。显然,$|T|=\binom{n}{k}$,而 A 中的 k 组合可以看作 $n-1$ 个元素的集合 $S\backslash\{a\}$ 中的 k 组合,因此,$|A|=\binom{n-1}{k}$,B 中的 k 组合可以看作把 a 添加到 $n-1$ 个元素的集合 $S\backslash\{a\}$ 中的 $k-1$ 组合中得到的,因此,$|B|=\binom{n-1}{k-1}$。结合以上事实有

$$\binom{n}{k}=\binom{n-1}{k}+\binom{n-1}{k-1}$$

利用 Pascal 公式和初始值 $\binom{n}{0}=1$ 和 $\binom{n}{n}=1(n\geqslant 0)$,可以容易地求出二项式系数,而不必借助于定理 4.1.3 中的公式。

定理 4.1.9(二项式展开) 对于 $n\geqslant 0$,有 $(a+b)^n=\sum_{k=0}^{n}\binom{n}{k}a^k b^{n-k}$。

证明:这个定理的一个证明是利用归纳法并应用 Pascal 公式。下面给出一个组合证法。因为

$$(a+b)^n=\underbrace{(a+b)(a+b)\cdots(a+b)}_{n\text{次}}$$

把这个乘积展开直到没有括号为止,由于每个因子$(a+b)$可以选择 a 或 b,其结果有 2^n 项且每项具有形式 $a^k b^{n-k}(k=0,1,\cdots,n)$。注意到,为了得到 $a^k b^{n-k}$,需从 n 个$(a+b)$中选出 k 个 a,在剩下的因子中选出 b,这种选法共有 $\binom{n}{k}$ 种。因此,$a^k b^{n-k}$ 在展开式中出现 $\binom{n}{k}$ 次。于是

$$(a+b)^n=\sum_{k=0}^{n}\binom{n}{k}a^k b^{n-k}$$

下面给出二项式展开式的一些应用。在定理 4.1.6 中,若取 $a=b=1$,得到

$$\binom{n}{0}+\binom{n}{1}+\cdots+\binom{n}{n}=2^n,\quad n\geqslant 0$$

若取 $a=1,b=-1$,得到

$$\binom{n}{0}-\binom{n}{1}+\cdots+(-1)^k\binom{n}{k}+\cdots+(-1)^n\binom{n}{n}=0,\quad n>0$$

由上式易得

$$\binom{n}{0}+\binom{n}{2}+\cdots=\binom{n}{1}+\binom{n}{3}+\cdots,\quad n>0$$

这一式子可以解释为：从 n 个对象中选取偶数个对象的方法数等于选取奇数个对象的方法数。

4.2　鸽巢原理、容斥原理及其应用

4.2.1　鸽巢原理

定理 4.2.1（鸽巢原理的简单形式）　如果把 $n+1$ 个物品放入 n 个盒子里，那么至少有一个盒子包含两个或两个以上的物品。

证明：反证。如果这 n 个盒子中的每一个至多包含一个物品，那么物品的总数至多是 n。而有 $n+1$ 个物品，所以至少有一个盒子包含至少两件物品。

例 4.2.1　给定 m 个整数 $a_1,a_2,\cdots,a_m$，必存在整数 k 和 l，$0\leqslant k<l\leqslant m$，使得 $a_{k+1}+a_{k+2}+\cdots+a_l$ 能够被 m 整除。

证明：考虑 m 个和

$$a_1,a_1+a_2,a_1+a_2+a_3,\cdots,a_1+a_2+a_3+\cdots+a_m$$

如果这些和中的有一个能被 m 整除，那么结论成立。因此可以假设这些和中的任一个都不能被 m 整除，那么它们除以 m 的余数只能为 $1,2,\cdots,m-1$。因为存在 m 个和而只有 $m-1$ 个余数，所以由鸽巢原理，必有两个和除以 m 的余数相同，即存在整数 k 和 l，$k<l$，使得 $a_1+a_2+\cdots+a_k$ 和 $a_1+a_2+\cdots+a_l$ 除以 m 有相同的余数，于是它们的差能被 m 整除。即 $a_{k+1}+a_{k+2}+\cdots+a_l$ 能被 m 整除。

对任意实数 x，令$\lfloor x\rfloor$表示不大于 x 的最大整数，称为 x 的弱取整，令$\lceil x\rceil$表示不小于 x 的最小整数，称为 x 的强取整。

定理 4.2.2（鸽巢原理的加强形式）　如果把 m 个物品放入 k 个盒子里，那么至少有一个盒子包含的物品数超过

$$\left\lfloor\frac{m-1}{k}\right\rfloor$$

证明：如果每个盒子里的物品数至多是 $\left\lfloor\frac{m-1}{k}\right\rfloor$，那么物品的总数至多是$k\left\lfloor\frac{m-1}{k}\right\rfloor$，而

$$k\left\lfloor\frac{m-1}{k}\right\rfloor\leqslant k\,\frac{m-1}{k}=m-1<m$$

矛盾，所以至少有一个盒子包含的物品数超过$\left\lfloor\frac{m-1}{k}\right\rfloor$。

例 4.2.2　如果一个房间里有 40 人，那么至少有几个人的生日在同一月份？

解：此问题相当于把 40 个物品放入 12 个盒子里，那么至少有一个盒子包含的物品数超过$\left\lfloor\frac{40-1}{12}\right\rfloor=3$。所以至少有 3 个人的生日在同一月份。

例 4.2.3 证明每个由 n^2+1 个实数构成的序列 $a_1,a_2,\cdots,a_{n^2+1}$，或者含有长度为 $n+1$ 的递增子序列，或者含有长度为 $n+1$ 的递减子序列。

证明：在证明之前先给出子序列的概念。如果 $a_1,a_2,\cdots,a_m$ 是一个序列，那么 $a_{i_1},a_{i_2},\cdots,a_{i_k}$ 就是一个子序列，其中 $1\leqslant i_1<i_2<\cdots<i_k\leqslant m$。若子序列 $a_{i_1},a_{i_2},\cdots,a_{i_k}$ 满足 $a_{i_1}\leqslant a_{i_2}\leqslant\cdots\leqslant a_{i_k}$ 则称为递增的，满足 $a_{i_1}\geqslant a_{i_2}\geqslant\cdots\geqslant a_{i_k}$ 则称为递减的。下面给出这个例子的证明。

假设不存在长度为 $n+1$ 的递增子序列，现在就来构造长度为 $n+1$ 的递减子序列。对每一个 $k=1,2,\cdots,n^2+1$，令 m_k 为从 a_k 开始的最长的递增子序列的长度。由假设知 $m_k\leqslant n$，而 $m_k\geqslant 1$ 对每一个 $k=1,2,\cdots,n^2+1$ 都成立，因此 $1\leqslant m_1,m_2,\cdots,m_{n^2+1}\leqslant n$。又$\left\lfloor\frac{n^2+1-1}{n}\right\rfloor=n$，则由定理 4.2.2 知，$m_1,m_2,\cdots,m_{n^2+1}$中至少有 $n+1$ 个数是相等的。令

$$m_{k_1}=m_{k_2}=\cdots=m_{k_{n+1}}$$

其中 $1\leqslant k_1<k_2<\cdots<k_{n+1}\leqslant n^2+1$。假设存在某个 $i(i=1,2,\cdots,n)$，使得 $a_{k_i}<a_{k_{i+1}}$。因为 $k_i<k_{i+1}$，可以把从 $a_{k_{i+1}}$ 开始的最长递增子序列的前面再加上一项 a_{k_i}，得到一个从 a_{k_i} 开始的递增子序列，其项数为 $m_{k_{i+1}}+1$，所以 $m_{k_i}>m_{k_{i+1}}$，与 $m_{k_i}=m_{k_{i+1}}$ 矛盾。因此 $a_{k_i}\geqslant a_{k_{i+1}}$，对于每个 $i=1,2,\cdots,n$ 都成立。于是得到一个长度为 $n+1$ 的递减子序列，即

$$a_{k_1}\geqslant a_{k_2}\geqslant\cdots\geqslant a_{k_{n+1}}$$

最后给出鸽巢原理的一般形式，即下面的定理 4.2.3。

定理 4.2.3 令 $q_1,q_2,\cdots,q_n$ 为正整数。如果将

$$q_1+q_2+\cdots+q_n-n+1$$

个物品放入 n 个盒子里，那么或者第一个盒子至少含有 q_1 个物品，或者第二个盒子至少含有 q_2 个物品，……，或者第 n 个盒子至少含有 q_n 个物品。

证明：将 $q_1+q_2+\cdots+q_n-n+1$ 个物品放入 n 个盒子里，如果对每一个 $i(i=1,2,\cdots,n)$，第 i 个盒子所含的物品少于 q_i，那么所有盒子的物品总数不超过

$$(q_1-1)+(q_2-1)+\cdots+(q_n-1)=q_1+q_2+\cdots+q_n-n$$

该数比所要分发的物品少 1。因此存在某个 $i(i=1,2,\cdots,n)$，使得第 i 个盒子里至少包含 q_i 个物品。

4.2.2 Ramsey 定理

下面叙述鸽巢原理的一个深刻且重要的推广，但不予以证明，其证明参见文献[3]。

定理 4.2.4 设 $q_1,q_2,\cdots,q_n,t$ 是正整数，且 $q_1\geqslant t,q_2\geqslant t,\cdots,q_n\geqslant t$，则存在一个正整数，从而存在最小的正整数 $N(q_1,q_2,\cdots,q_n;t)$，它仅依赖于 $q_1,q_2,\cdots,q_n$ 和 t，并

具有下面的性质：如果 $m \geqslant N(q_1,q_2,\cdots,q_n;t)$ 且 S 是 m 个元素的集合，把 S 的 t 元子集分布在 n 个盒子里，那么或者有 q_1 个元素使它们的全部 t 元子集都分布在第一个盒子里，或者有 q_2 个元素使它们的全部 t 元子集都分布在第二个盒子里，……，或者有 q_n 个元素使它们的全部 t 元子集都分布在第 n 个盒子里。

数 $N(q_1,q_2,\cdots,q_n;t)$ 称为 **Ramsey 数**。确定 Ramsey 数是一个困难的问题，关于这些数有很少的研究结果。已知 $N(3,3;2)=6$，对于这种情况，Ramsey 定理可以叙述如下，这也是 Ramsey 定理最简单的情况。

定理 4.2.5　假设在 6 个人中间每两个人或者是朋友或者是敌人。那么或者有 3 个人互为朋友，或者有 3 个人互为敌人。

证明：设 a 是 6 个人中间的任一人，现把剩下的 5 人分成两个集合，A 表示与 a 是朋友的人所成集合，B 表示与 a 是敌人的人所成集合。因 $\left\lceil \frac{5-1}{2} \right\rceil = 2$（在定理 4.2.2 中取 $m=5, k=2$），则由定理 4.2.2，在剩下的 5 人中，或者有 3 人或更多的人是 a 的朋友，或者有 3 人或更多的人是 a 的敌人。首先假设 b、c、d 是 a 的朋友，若 b、c、d 中有两人是朋友，那么这两人与 a 组成互为朋友的 3 人组；若 b、c、d 中任两个人都不是朋友，那么 b、c、d 就形成一个互为敌人的 3 人组。b、c、d 是 a 的敌人时可类似证明。

Ramsey 定理在实际中有许多有趣的应用，有兴趣的读者可以参见文献[7]。

4.2.3　容斥原理

下面介绍另一个基本的计数工具——容斥原理。先通过一个例子来介绍这一原理的基本思想。

例 4.2.4　假设有 25 名求职者，其中 12 人有计算机编程技术，7 人有计算机硬件技术，2 人既有计算机编程技术又有计算机硬件技术。问这些人中有多少人既没有计算机编程技术也没有计算机硬件技术？

解：为了知道有多少人既没有计算机编程技术也没有计算机硬件技术，就要从 25 人中减去有编程技术的 12 人，再减去有硬件技术的 7 人。但这样就把既有编程技术又有硬件技术的 2 人减去了两次，因此必须把这两人再加回去。于是既没有计算机编程技术也没有计算机硬件技术的人数为

$$25 - 12 - 7 + 2 = 8$$

现在来扩展上述的推理。

令 S 是物体的有限集，$P_1,P_2,\cdots,P_m$ 是 S 的物体所涉及的 m 个性质。又令

$$A_i = \{x : x \text{ 在 } S \text{ 内且 } x \text{ 具有性质 } P_i\}, \quad i = 1,2,\cdots,m$$

是 S 的具有性质 P_i 的物体构成的子集。因此 $A_i \cap A_j$ 是同时具有性质 P_i 和 P_j 的物体构成的子集，$A_i \cap A_j \cap A_k$ 是同时具有性质 P_i、P_j 和 P_k 的物体构成的子集等。哪个性质都不具有的物体构成的子集是 $\overline{A}_1 \cap \overline{A}_2 \cap \cdots \cap \overline{A}_m$。

定理 4.2.6（容斥原理）　有限集 S 的不具有性质 $P_1,P_2,\cdots,P_m$ 的物体的个数为

$$
\begin{aligned}
&|\bar{A}_1 \cap \bar{A}_2 \cap \cdots \cap \bar{A}_m| \\
=&|S| - \sum |A_i| + \sum |A_i \cap A_j| - \sum |A_i \cap A_j \cap A_k| \\
&+ \cdots + (-1)^m |A_1 \cap A_2 \cap \cdots \cap A_m|
\end{aligned} \tag{4.1}
$$

其中，$\sum |A_i|$ 是对$\{1,2,\cdots,m\}$中的所有整数 i 求和；$\sum |A_i \cap A_j|$ 是对$\{1,2,\cdots,m\}$中的所有2组合$\{i,j\}$求和；$\sum |A_i \cap A_j \cap A_k|$ 是对$\{1,2,\cdots,m\}$中的所有3组合$\{i,j,k\}$求和等。

证明：因为等式左边表示 S 中不具有性质 $P_1,P_2,\cdots,P_m$ 的物体的计数，所以只需要证明 S 中不具有性质 $P_1,P_2,\cdots,P_m$ 的每一个物体在等式右边正好被计数1次，而至少具有其中一个性质的每一个物体在等式右边正好被计数0次。

首先假设物体 x 不具有性质 $P_1,P_2,\cdots,P_m$，那么它在$|S|$中被计数1次，但在 $\sum |A_i|$、$\sum |A_i \cap A_j|$、$\sum |A_i \cap A_j \cap A_k|$、…、$|A_1 \cap A_2 \cap \cdots \cap A_m|$ 中却不被计数。因此它在等式右边正好被计数1次。

现假设物体 x 至少具有性质 $P_1,P_2,\cdots,P_m$ 中的一个，不妨假设 x 恰好具有 $P_1,P_2,\cdots,P_m$ 中的 n 个，其中 $n\geqslant 1$，则 x 在$|S|$中被计数1次，$1=\binom{n}{0}$。由于 x 恰好具有 n 个性质，因此它在 $\sum |A_i|$ 中恰好被计数 n 次，$n=\binom{n}{1}$。又由于从 n 个性质中取出两个性质的取法有$\binom{n}{2}$种，因此它在$\sum |A_i \cap A_j|$中正好被计数$\binom{n}{2}$次。同理，x 在$\sum |A_i \cap A_j \cap A_k|$中正好被计数$\binom{n}{3}$次等。于是，$x$ 在等式(4.1)右边正好被计数

$$
\binom{n}{0} - \binom{n}{1} + \binom{n}{2} - \binom{n}{3} + \cdots + (-1)^m \binom{n}{m}
$$

次。因为 $n\leqslant m$，所以它等于

$$
\binom{n}{0} - \binom{n}{1} + \binom{n}{2} - \binom{n}{3} + \cdots + (-1)^n \binom{n}{n} = (1-1)^n = 0
$$

因此，如果 x 至少具有性质 $P_1,P_2,\cdots,P_m$ 中的一个，它在等式(4.1)右边正好被计数0次。于是等式(4.1)成立。

推论 4.2.1 有限集 S 中至少具有性质 $P_1,P_2,\cdots,P_m$ 之一的物体的个数为

$$
\begin{aligned}
&|A_1 \cup A_2 \cup \cdots \cup A_m| \\
=&\sum |A_i| - \sum |A_i \cap A_j| + \sum |A_i \cap A_j \cap A_k| \\
&+ \cdots + (-1)^{m+1} |A_1 \cap A_2 \cap \cdots \cap A_m|
\end{aligned}
$$

例 4.2.5 求从1～1000不能被5、6、8整除的整数个数。

解：为方便起见用lcm表示整数的最小公倍数。令 S 表示前1000个正整数组成的集合，P_1 表示具有能被5整除的性质，P_2 表示具有能被6整除的性质，P_3 表示具有能被8整除的性质，$A_i(i=1,2,3)$表示 S 中具有性质 P_i 的整数组成的集合。于

是本题就是要求 $|\overline{A}_1\cap\overline{A}_2\cap\overline{A}_3|$ 等于多少。

显然，$|A_1|=\left\lfloor\frac{1000}{5}\right\rfloor=200$，$|A_2|=\left\lfloor\frac{1000}{6}\right\rfloor=166$，$|A_3|=\left\lfloor\frac{1000}{8}\right\rfloor=125$。

而集合 $A_1\cap A_2$ 中的整数可同时被 5 和 6 整除。但一个整数同时被 5 和 6 整除当且仅当它能被 $\mathrm{lcm}\{5,6\}=30$ 整除，因此

$$|A_1\cap A_2|=\left\lfloor\frac{1000}{\mathrm{lcm}\{5,6\}}\right\rfloor=\left\lfloor\frac{1000}{30}\right\rfloor=33$$

$$|A_1\cap A_3|=\left\lfloor\frac{1000}{\mathrm{lcm}\{5,8\}}\right\rfloor=\left\lfloor\frac{1000}{40}\right\rfloor=25$$

$$|A_2\cap A_3|=\left\lfloor\frac{1000}{\mathrm{lcm}\{6,8\}}\right\rfloor=\left\lfloor\frac{1000}{24}\right\rfloor=41$$

同理

$$|A_1\cap A_2\cap A_3|=\left\lfloor\frac{1000}{\mathrm{lcm}\{5,6,8\}}\right\rfloor=\left\lfloor\frac{1000}{120}\right\rfloor=8$$

由容斥原理从 1～1000 不能被 5、6、8 整除的整数个数为

$$\begin{aligned}|\overline{A}_1\cap\overline{A}_2\cap\overline{A}_3|&=1000-(200+166+125)+(33+25+41)-8\\&=600\end{aligned}$$

例 4.2.6　称两个整数是互素的，如果它们没有大于 1 的公约数。问 1～1000 之间有多少个整数与 100 互素？

解：因为 $100=2^2\times5^2$，整除 100 的素数只有 2 和 5，所以只要找出 1～1000 之间不能被 2、5 整除的整数的个数即可。利用容斥原理容易算出这个数为 400。

4.2.4　重复组合

在 4.1.3 小节和 4.1.5 小节中，已经证明 n 个不同元素的集合的 r 组合的个数为 $\binom{n}{r}$，具有 k 种不同元素且每种元素都有无限重复数的重集的 r 组合的个数为 $\binom{r+k-1}{r}$。本节利用容斥原理给出一种方法来求出对于重数没有任何限制的重集的 r 组合的个数。下面通过一个特殊的例子来讨论，但对一般情况这个方法也是有效的。

例 4.2.7　确定重集 $T=\{3\cdot a,4\cdot b,5\cdot c\}$ 的 10 组合的个数。

解：做重集 $T^*=\{\infty\cdot a,\infty\cdot b,\infty\cdot c\}$，令 S 表示 T^* 的所有 10 组合组成的集合，P_1 是 T^* 的 10 组合具有多于 3 个 a 的性质，P_2 是 T^* 的 10 组合具有多于 4 个 b 的性质，P_3 是 T^* 的 10 组合具有多于 5 个 c 的性质。同样，令 $A_i(i=1,2,3)$ 表示由 T^* 中具有性质 P_i 的所有 10 组合构成的集合。因此，要确定 T 的 10 组合的个数，只要求出 $|\overline{A}_1\cap\overline{A}_2\cap\overline{A}_3|$ 等于多少即可。由容斥原理

$$\begin{aligned}&|\overline{A}_1\cap\overline{A}_2\cap\overline{A}_3|\\&=|S|-(|A_1|+|A_2|+|A_3|)+(|A_1\cap A_2|+|A_1\cap A_3|\\&\quad+|A_2\cap A_3|)-|A_1\cap A_2\cap A_3|\end{aligned}$$

由定理 4.1.7

$$|S|=\binom{10+3-1}{10}=\binom{12}{10}=66$$

已知 A_1 表示 a 至少出现 4 次的 T^* 的所有 10 组合构成的集合。如果任取 A_1 中的一个 10 组合并从中去掉 4 个 a，那么就得到 T^* 的一个 6 组合。反之，如果任取 T^* 的一个 6 组合并往里加入 4 个 a，就得到 T^* 的一个 a 至少出现 4 次的 10 组合。因此，A_1 中的 10 组合的个数就等于 T^* 的 6 组合的个数。于是

$$|A_1|=\binom{6+3-1}{6}=\binom{8}{6}=28$$

同理，A_2 中的 10 组合的个数就等于 T^* 的 5 组合的个数，A_3 中的 10 组合的个数就等于 T^* 的 4 组合的个数。因此

$$|A_2|=\binom{5+3-1}{5}=\binom{7}{5}=21$$

$$|A_3|=\binom{4+3-1}{4}=\binom{6}{4}=15$$

类似地，已知 $A_1\cap A_2$ 表示 a 至少出现 4 次且 b 至少出现 5 次的 T^* 的所有 10 组合构成的集合。如果任取 $A_1\cap A_2$ 中的一个 10 组合并从中去掉 4 个 a 和 5 个 b，那么就得到 T^* 的一个 1 组合。反之，如果任取 T^* 的一个 1 组合并往里加入 4 个 a 和 5 个 b，就得到 T^* 的一个 a 至少出现 4 次且 b 至少出现 5 次的 10 组合。因此，$A_1\cap A_2$ 中的 10 组合的个数就等于 T^* 的 1 组合的个数。于是

$$|A_1\cap A_2|=\binom{1+3-1}{1}=\binom{3}{1}=3$$

同理，$A_1\cap A_3$ 中的 10 组合的个数就等于 T^* 的 0 组合的个数，并且在 $A_2\cap A_3$ 中没有 10 组合。于是

$$|A_1\cap A_3|=\binom{0+3-1}{0}=\binom{2}{0}=1$$

$$|A_2\cap A_3|=0$$

从而

$$|A_1\cap A_2\cap A_3|=0$$

由容斥原理得

$$|\overline{A}_1\cap\overline{A}_2\cap\overline{A}_3|=66-(28+21+15)+(3+1+0)-0=6$$

所以重集 $T=\{3\cdot a,4\cdot b,5\cdot c\}$的 10 组合的个数等于 6。

4.2.5 错位排列

集合$\{1,2,\cdots,n\}$的一个排列 $i_1,i_2,\cdots,i_n$ 称为它的一个**错位排列**，如果 $i_1\neq 1$，$i_2\neq 2,\cdots,i_n\neq n$。也就是说，$\{1,2,\cdots,n\}$的一个错位排列就是$\{1,2,\cdots,n\}$的每个元素都不在其自然位置上的一个排列。用 D_n 表示$\{1,2,\cdots,n\}$的错位排列的个数，简称**错位数**。

定理 4.2.7　对于 $n\geqslant 1$，

$$D_n=n!\left(1-\frac{1}{1!}+\frac{1}{2!}-\frac{1}{3!}+\cdots+(-1)^n\frac{1}{n!}\right)$$

证明：令 S 是$\{1,2,\cdots,n\}$的所有排列组成的集合，则$|S|=n!$。对于 $j=1,2,\cdots,n$，令 P_j 表示排列中 j 位于它的自然位置上这一性质，A_j 表示具有性质 P_j 的$\{1,2,\cdots,n\}$的排列组成的集合。于是$\{1,2,\cdots,n\}$的错位排列恰好就是 $\overline{A}_1\cap\overline{A}_2\cap\cdots\cap\overline{A}_n$ 中的那些排列，也就是说，$D_n=|\overline{A}_1\cap\overline{A}_2\cap\cdots\cap\overline{A}_n|$。下面利用容斥原理来计算 D_n。

A_1 中的排列就是形如 $1i_2\cdots i_n$ 的那些排列，其中 $i_2\cdots i_n$ 是$\{2,3,\cdots,n\}$的一个排列，所以$|A_1|=(n-1)!$。同理可知

$$|A_j|=(n-1)!\quad j=1,2,\cdots,n$$

$A_1\cap A_2$ 中的排列就是形如 $12i_3\cdots i_n$ 的那些排列，其中 $i_3\cdots i_n$ 是$\{3,4,\cdots,n\}$的一个排列，所以$|A_1\cap A_2|=(n-2)!$。同理可知，对于$\{1,2,\cdots,n\}$的任意一个 2 组合$\{i,j\}$有

$$|A_i\cap A_j|=(n-2)!$$

完全类似地，对于$\{1,2,\cdots,n\}$的任意一个 k 组合$\{i_1,i_2,\cdots,i_k\}$有

$$|A_{i_1}\cap A_{i_2}\cap\cdots\cap A_{i_k}|=(n-k)!$$

因为$\{1,2,\cdots,n\}$的 k 组合有$\binom{n}{k}$个，利用容斥原理有

$$\begin{aligned}D_n&=n!-\binom{n}{1}(n-1)!+\binom{n}{2}(n-2)!\\&\quad-\binom{n}{3}(n-3)!+\cdots+(-1)^n\binom{n}{n}(n-n)!\\&=n!-\frac{n!}{1!}+\frac{n!}{2!}-\frac{n!}{3!}+\cdots+(-1)^n\frac{n!}{n!}\\&=n!\left(1-\frac{1}{1!}+\frac{1}{2!}-\frac{1}{3!}+\cdots+(-1)^n\frac{1}{n!}\right)\end{aligned}$$

例 4.2.8　有 n 个人参加一个晚会，每人寄存一顶帽子，晚会后每人随便拿回其中的一顶，求：

(1) 没有一个人拿回自己原来的帽子的概率；

(2) 至少有一个人拿回自己原来的帽子的概率；

(3) 至少有两个人拿回自己原来的帽子的概率。

解：(1) 每人随便拿一顶帽子相当于 n 顶帽子的一个排列，其个数为 $n!$。而没有一个人拿回自己的帽子恰好是一个错位排列，其个数为 D_n。所以没有一个人拿回自己的帽子的概率为

$$\begin{aligned}\frac{D_n}{n!}&=\frac{n!\left(1-\frac{1}{1!}+\frac{1}{2!}-\frac{1}{3!}+\cdots+(-1)^n\frac{1}{n!}\right)}{n!}\\&=1-\frac{1}{1!}+\frac{1}{2!}-\frac{1}{3!}+\cdots+(-1)^n\frac{1}{n!}\end{aligned}$$

$$\approx e^{-1} \approx 37\%$$

这是因为 $e^{-1}=1-\frac{1}{1!}+\frac{1}{2!}-\frac{1}{3!}+\cdots+(-1)^n\frac{1}{n!}+\cdots$。

（2）至少有一个人拿回自己原来的帽子的概率约为

$$1-e^{-1} \approx 63\%$$

（3）因为没有一个人拿回自己的帽子的情况有 D_n 种，恰有一个人拿回自己的帽子的情况有 $\binom{n}{1}D_{n-1}$ 种，因此至少有两个人拿回自己原来的帽子的情况有

$$\begin{aligned}
& n!-\left(D_n+\binom{n}{1}D_{n-1}\right) \\
=& n!-\left(n!\left(1-\frac{1}{1!}+\frac{1}{2!}-\frac{1}{3!}+\cdots+(-1)^n\frac{1}{n!}\right)\right. \\
& \left.+n(n-1)!\left(1-\frac{1}{1!}+\frac{1}{2!}-\frac{1}{3!}+\cdots+(-1)^{n-1}\frac{1}{(n-1)!}\right)\right) \\
=& n!-\left(2n!\left(1-\frac{1}{1!}+\frac{1}{2!}-\frac{1}{3!}+\cdots+(-1)^n\frac{1}{n!}\right)-(-1)^n\frac{n!}{n!}\right) \\
=& n!-(2D_n-(-1)^n)
\end{aligned}$$

其概率为

$$\frac{n!-(2D_n-(-1)^n)}{n!}=1-2\frac{D_n}{n!}+\frac{(-1)^n}{n!}\approx 1-2e^{-1}\approx 26\%$$

从此例中可以看到一个有趣的现象：由交错级数的基本事实，e^{-1} 和 $\frac{D_n}{n!}$ 之差的绝对值小于 $\frac{1}{(n+1)!}$，而通过计算知，当 $n\geqslant 7$ 时，e^{-1} 和 $\frac{D_n}{n!}$ 至少有 3 位小数相同。从实际的观点来看，$n\geqslant 7$ 时，e^{-1} 和 $\frac{D_n}{n!}$ 是一样的。也就是说，只要参加晚会的人数不少于 7 人，那么没有一个人拿回自己原来的帽子的概率大约都是 $e^{-1}\approx 37\%$。

关于错位数 D_n，还有一些有趣的关系。利用定理 4.2.7 易得下面的递推关系：

$$D_n=(n-1)(D_{n-2}+D_{n-1}) \quad n=3,4,5,\cdots \tag{4.2}$$

由初始条件：$D_1=0, D_2=1$，可以计算相应于任何整数 n 的 D_n。例如

$$\begin{aligned}
D_3 &= (3-1)(D_{3-2}+D_{3-1})=2(0+1)=2 \\
D_4 &= (4-1)(D_{4-2}+D_{4-1})=3(1+2)=9 \\
D_5 &= (5-1)(D_{5-2}+D_{5-1})=4(2+9)=44
\end{aligned}$$

把式(4.2)变形有，当 $n\geqslant 3$ 时

$$\begin{aligned}
D_n-nD_{n-1} &= -[D_{n-1}-(n-1)D_{n-2}] \\
&= (-1)^2[D_{n-2}-(n-2)D_{n-3}] \\
&= (-1)^3[D_{n-3}-(n-3)D_{n-4}] \\
&\;\;\vdots \\
&= (-1)^{n-2}(D_2-2D_1)
\end{aligned}$$

$$=(-1)^{n-2}$$

于是得到关于错位数 D_n 的一个更简单的递推关系：

$$D_n = nD_{n-1} + (-1)^{n-2}$$

或等价地

$$D_n = nD_{n-1} + (-1)^n,\quad n = 2,3,4,\cdots \tag{4.3}$$

需要注意的是，上述验证只对 $n=3,4,5,\cdots$ 进行，而当 $n=2$ 时，容易证明式(4.3)也成立。

4.2.6　其他禁位问题

上一节已经讨论了如何计算 $\{1,2,\cdots,n\}$ 的且第 j 个位置上禁止出现 $j(j=1,2,\cdots,n)$ 的排列的个数问题。本节将讨论 $\{1,2,\cdots,n\}$ 的某些相对禁用位置的排列的计数问题。具体地说就是：给定一个正整数 n，如何去计算模式 $12,23,\cdots,(n-1)n$ 中任何一个都不出现的 $\{1,2,\cdots,n\}$ 的排列的个数 Q_n。

定理 4.2.8　对于 $n\geqslant 1$，有

$$Q_n = n! - \binom{n-1}{1}(n-1)! + \binom{n-1}{2}(n-2)! - \binom{n-1}{3}(n-3)! + \cdots + (-1)^{n-1}\binom{n-1}{n-1}1!$$

证明：令 S 是 $\{1,2,\cdots,n\}$ 的所有排列组成的集合，则 $|S|=n!$。对于 $j=1,2,\cdots,n-1$，令 P_j 表示排列中出现模式 $j(j+1)$ 这一性质，A_j 表示具有性质 P_j 的 $\{1,2,\cdots,n\}$ 的排列组成的集合。于是 $\{1,2,\cdots,n\}$ 中不出现模式 $12,23,\cdots,(n-1)n$ 的排列恰好就是 $\bar{A}_1\cap\bar{A}_2\cap\cdots\cap\bar{A}_n$ 中的那些排列，也就是说，$Q_n=|\bar{A}_1\cap\bar{A}_2\cap\cdots\cap\bar{A}_n|$。同样利用容斥原理来计算 Q_n。

首先计算 A_1 中排列的个数。一个排列在 A_1 中当且仅当模式 12 出现在此排列中，所以 A_1 中一个排列可以看成 $n-1$ 个符号 $\{12,3,\cdots,n\}$ 的一个排列。于是 $|A_1|=(n-1)!$，一般地

$$|A_j| = (n-1)!,\quad j = 1,2,\cdots,n-1$$

在集合 $A_1,A_2,\cdots,A_{n-1}$ 中任取两个集合，这两个集合只有两种情况，一种是两者的下标相邻，比如 A_1,A_2，另一种是两者的下标不相邻，比如 A_1,A_3。$A_1\cap A_2$ 中的排列出现模式 12 和 23，这种排列可以看成 $n-2$ 个符号 $\{12,34,\cdots,n\}$ 的一个排列，于是 $|A_1\cap A_2|=(n-2)!$。而 $A_1\cap A_3$ 中的排列出现模式 12 和 34，这种排列可以看成 $n-2$ 个符号 $\{12,34,\cdots,n\}$ 的一个排列，于是 $|A_1\cap A_3|=(n-2)!$。一般地

$$|A_i \cap A_j| = (n-2)!$$

对于 $\{1,2,\cdots,n\}$ 的每一个 2 组合 $\{i,j\}$ 成立。更一般地，包含模式 $12,23,\cdots,(n-1)n$ 中的 $k(k\leqslant n-1)$ 个特定模式的排列可以看成 $n-k$ 个符号的排列，这样，对于 $\{1,2,\cdots,n-1\}$ 的任意一个 k 组合 $\{i_1,i_2,\cdots i_k\}$ 有

$$|A_{i_1} \cap A_{i_2} \cap \cdots \cap A_{i_k}| = (n-k)!$$

因为$\{1,2,\cdots,n-1\}$的k组合有$\binom{n-1}{k}$个，利用容斥原理有

$$Q_n = n! - \binom{n-1}{1}(n-1)! + \binom{n-1}{2}(n-2)! - \binom{n-1}{3}(n-3)! + \cdots + (-1)^{n-1}\binom{n-1}{n-1}1!$$

例 4.2.9 n个小朋友排成一列纵队散步，假定每一位小朋友只能看到他前面的一位小朋友（第一位除外）。问有多少种变换队形的方法，使得每一个小朋友不再看到当初他看到的小朋友？

解：把n个小朋友按初始队形顺序编号为$1,2,\cdots,n$，则问题就等价于求$\{1,2,\cdots,n\}$的排列中没有模式$12,23,\cdots,(n-1)n$出现的排列数，所以答案是Q_n。

4.3 区组设计和拉丁方

4.3.1 区组设计

首先用一个统计分析中试验设计的简化例子来说明区组设计的含义。

例 4.3.1 假设一个产品有 7 种样品需要测试，制造商计划随机请一些消费者来比较这些样品。一种方法是让每一位参加试验的消费者进行全面的测试：比较所有 7 种样品。但这样花费的时间和精力会比较多，不是每个消费者都愿意的，因此决定让每一位参加试验的消费者进行非全面的测试，比较其中的某些样品。现制造商要求每个人比较 3 件样品。为了能够得出基于结果的统计分析的有意义的结论，测试要具有这样的性质：7 种样品中的每一对样品恰被一人比较。问能否设计出这样一种测试试验？

解：把这 7 种样品标记为 0,1,2,3,4,5,6。这 7 种样品总共可以配成$\binom{7}{2}=21$对。每个测试人得到 3 种样品，进行$\binom{3}{2}=3$次比较。由于每一对恰被比较 1 次，故测试者的人数应为$\frac{21}{3}=7$人。幸好这个商是一个整数，否则就不能用所给的限制设计出一个试验。

现在要寻找的是这 7 种样品的 7 个子集$B_1,B_2,\cdots,B_7$（每个测试者一个），称之为**区组**（block），它们具有性质：每一对样品恰好在一个区组内。下面是这样的一个区组：

$$B_1=\{0,1,3\},\quad B_2=\{1,2,4\},\quad B_3=\{2,3,5\},\quad B_4=\{3,4,6\},$$
$$B_5=\{0,4,5\},\quad B_6=\{1,5,6\},\quad B_7=\{0,2,6\}$$

描述该试验设计的另一种方式是利用下面的列表给出。在该列表中，列对应于样品，行对应于区组，如果样品j属于区组B_i，那么i行j列交叉处为 1，否则为 0。列表如下：

	0	1	2	3	4	5	6
B_1	1	1	0	1	0	0	0
B_2	0	1	1	0	1	0	0
B_3	0	0	1	1	0	1	0
B_4	0	0	0	1	1	0	1
B_5	1	0	0	0	1	1	0
B_6	0	1	0	0	0	1	1
B_7	1	0	1	0	0	0	1

从表中可以看出，每一行含有 3 个 1，反映了每个区组含有 3 个样品这一事实，每两列恰在且只在一行上同时含有 1，反映了每一对样品恰好在同一个区组中这一事实。

可以把此表利用矩阵的形式表示如下：

$$\begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{pmatrix}$$

称之为试验设计的**关联矩阵**。

现在来定义一些术语并讨论区组设计的一些初等性质。

定义 4.3.1　设 V 是含有 $v(v\geqslant 2)$ 个元素(有时称之为样品)的集合，$B=\{B_1, B_2, \cdots, B_b\}$，其中 $B_1, B_2, \cdots, B_b$ 是 V 的子集(称之为区组)。如果 B 满足下面的条件：

(1) $B_1, B_2, \cdots, B_b$ 都含有 k 个元素，$k>0$；

(2) V 的每一个元素恰好出现在 r 个区组中，$r>0$；

(3) V 的每一对元素恰好出现在 λ 个区组中，$\lambda>0$。

则称 B 是 V 的一个**平衡区组设计**，数 λ 称为**设计指数**。

满足 $k<v$ 的平衡区组设计称为**平衡不完全区组设计**(Balanced Incomplete Block Design，BIBD)，有时也称为($\boldsymbol{b}, \boldsymbol{v}, \boldsymbol{r}, \boldsymbol{k}, \boldsymbol{\lambda}$)**设计**。

应该注意到，如果 $k=v$，即每一个区组 $B_i(i=1,2,\cdots,b)$ 都等于 V，那么定义 4.3.1 中的 3 个条件自然满足，且 $k=v, r=b, \lambda=b$，这时则称 B 为**完全区组设计**。在例 4.3.1 中，一个完全区组设计对应于使每个人都要比较每一对样品的试验。从组合学的观点来看，完全区组设计是平凡的。因此下面的讨论中，除特别声明外，都假设 $k<v$。

由上定义可知，例 4.3.1 是一个(7,7,3,3,1)设计。

令 B 是 V 上的一个 BIBD。如上例所看到的，把 B 与一个关联矩阵 $\mathbf{A}$ 相联系。矩阵 $\mathbf{A}$ 是由 0 和 1 组成的 $b\times v$ 矩阵，它的行对应每一个区组 $B_1,B_2,\cdots,B_b$，它的列对应 V 中的每一个样品 $x_1,x_2,\cdots,x_v$，位于 i 行 j 列交叉处的元素 $a_{ij}(i=1,2,\cdots,b;j=1,2,\cdots,v)$定义如下：

$$a_{ij}=\begin{cases}1 & 若\ x_j\ 在\ B_i\ 中\\ 0 & 若\ x_j\ 不在\ B_i\ 中\end{cases}$$

尽管关联矩阵依赖于排列区组的顺序和排列样品的顺序，还是称它为 B 的关联矩阵。关联矩阵包含了 BIBD 的全部信息。

对于 $v\geqslant 2$，任意 $b\times v$ 的 0 和 1 矩阵是(b,v,r,k,λ)设计的关联矩阵，其中 $b,v,r,k,\lambda>0$，当且仅当下面条件成立：

(1) 每一列中 1 的数目相同，都是 r 个，且 $r>0$；

(2) 每一行中 1 的数目相同，都是 k 个，且 $k>0$；

(3) 每两列中同时为 1 的行数相同，都是 λ 个，且 $\lambda>0$。

一个 BIBD 包含 5 个参数，即 b,v,r,k,λ，其中

b：区组个数；

v：样品个数；

r：包含每个样品的区组的个数；

k：在每个区组中样品的个数；

λ：包含每对样品的区组的个数。

下面给出这些参数必须满足的条件。

定理 4.3.1 在一个(b,v,r,k,λ)设计中，有

$$r(k-1)=\lambda(v-1) \tag{4.4}$$

证明：令 x_i 为任一样品，并设 x_i 含于 r 个区组

$$B_{i_1},B_{i_2},\cdots,B_{i_r} \tag{4.5}$$

中。由于每一个区组含有 k 个元素，因此这些区组中的每一个都含有 $k-1$ 个异于 x_i 的样品。现在考虑 $v-1$ 个元素对$\{x_i,y\}$，其中 y 是异于 x_i 的样品。对于每一个这样的元素对，现在计算同时包含这两个样品的区组数(这些区组一定包含在式(4.5)中的这 r 个区组中，因为它们是包含 x_i 的全部区组)。由条件知，每一对区组均含于 λ 个区组中，相加后得到

$$\lambda(v-1)$$

另一方面，式(4.5)中的每个组均含有 $k-1$ 个包含 x_i 的元素对，相加后得到

$$(k-1)r$$

因这两个数相等，于是

$$(k-1)r=\lambda(v-1)$$

由定理 4.3.1 知，一个 BIBD 中包含的 5 个参数 b、v、r、k、λ 不是独立的，并且从它的证明中可以看出，定义 4.3.1 中的 3 个条件也不是独立的，由条件(1)、(3)可以

推出条件(2)。

推论 4.3.1　在 BIBD 中,有

$$bk = vr \tag{4.6}$$

证明:已经观察到,通过行来计数,BIBD 关联矩阵 $\mathbf{A}$ 中 1 的个数为 bk。由定理 4.3.1 知,$\mathbf{A}$ 的每一列含有 r 个 1,因此通过列来计数,$\mathbf{A}$ 中 1 的个数为 vr,两数相等,所以得到

$$bk = vr$$

推论 4.3.2　在 BIBD 中,有

$$\lambda < r$$

证明:根据定义,在 BIBD 中,$k<v$,从而 $k-1<v-1$,由定理 4.3.1 知 $\lambda<r$。

例 4.3.2　是否存在参数为 $b=12,k=4,v=16$ 及 $r=3$ 的 BIBD?

解:由定理 4.3.1,如果存在这样的设计,那么它的指数 λ 应满足

$$\lambda = \frac{r(k-1)}{v-1} = \frac{3\times(4-1)}{16-1} = \frac{9}{15}$$

因为它不是整数,所以不可能存在所给定的 4 个参数的 BIBD。

定理 4.3.2(Fisher 不等式)　在 BIBD 中,有

$$b \geqslant v \tag{4.7}$$

证明:令 $\mathbf{A}$ 为 BIBD 的 $b\times v$ 关联矩阵。由于每一个样品在 r 个区组中,每一对样品在 λ 个区组中,记 $\mathbf{A}$ 的转置矩阵为 $\mathbf{A}^{\mathrm{T}}$,因此 $\mathbf{A}^{\mathrm{T}}\mathbf{A}$ 为 $v\times v$ 方阵,且主对角线上的元素均为 r,主对角线以外的元素均为 λ,即

$$\mathbf{A}^{\mathrm{T}}\mathbf{A} = \begin{pmatrix} r & \lambda & \cdots & \lambda \\ \lambda & r & \cdots & \lambda \\ \vdots & \vdots & \ddots & \vdots \\ \lambda & \lambda & \cdots & r \end{pmatrix}$$

由推论 4.3.2 知 $\lambda<r$,利用线性代数的知识容易计算,$\mathbf{A}^{\mathrm{T}}\mathbf{A}$ 的行列式

$$|\mathbf{A}^{\mathrm{T}}\mathbf{A}| = (r-\lambda)^{v-1}(r+(v-1)\lambda) \neq 0$$

从而该矩阵是可逆的,其秩为 v,因此 $\mathbf{A}$ 的秩至少是 v。由于 $\mathbf{A}$ 是 b 行 v 列矩阵,于是必有 $b\geqslant v$。

使式(4.7)中等号成立的 BIBD,即区组个数 b 等于样品个数 v 的 BIBD 称为是**对称的**,简称 SBIBD。由于在 BIBD 中满足 $bk=vr$,于是在 SBIBD 中有 $k=r$。由式(4.4)知,SBIBD 的指数 λ 为

$$\lambda = \frac{k(k-1)}{v-1} \tag{4.8}$$

因此,SBIBD 的参数为:

b:区组个数;

v:样品个数;

k:在每个区组中样品的个数;

r：包含每个样品的区组的个数；

λ：包含每一对样品的区组的个数，其中 λ 由式(4.8)给出。

前面的例 4.3.1 就是一个 SBIBD。

现在讨论构造 SBIBD 的方法，该方法要用到整数 mod n 的运算。在这种方法中，样品是 Z_n 中的整数，为了与我们的记法一致，这里使用 v 而不使用 n。

令 $v \geqslant 2$ 为一整数，考虑 mod v 的整数集：

$$Z_v = \{0,1,2,\cdots,v-1\}$$

它的加法和乘法用通常的记号＋和×表示。令

$$B = \{i_1, i_2, \cdots, i_k\}$$

是 Z_v 的一个 k 元子集。对于 Z_v 中的任意一个整数 j ($j=0,1,2,\cdots,v-1$)，定义

$$B+j = \{i_1+j, i_2+j, \cdots, i_k+j\}$$

即 B 的每一个整数以 mod v 的方式加上整数 j。显然，$B+j$ 仍是 Z_v 的子集，并且是 Z_v 的 k 元子集。这是因为，如果 $i_p+j=i_q+j$（在 Z_v 中），两边消去 j，得到 $i_p=i_q$。这样就得到 Z_v 的 v 个子集

$$B = B+0, B+1, B+2, \cdots, B+v-1$$

称为从区组 B 发展起来的区组，而 B 叫做**初始值区组**(starter block)。

例 4.3.3 令 $v=7$，考虑 $Z_7=\{0,1,2,3,4,5,6\}$。设初始值区组为 $B=\{0,1,3\}$。于是

$$B+0=\{0,1,3\},\quad B+1=\{1,2,4\},\quad B+2=\{2,3,5\},$$
$$B+3=\{3,4,6\},\quad B+4=\{4,5,0\},\quad B+5=\{5,6,1\},$$
$$B+6=\{6,0,2\}$$

这是一个 BIBD，与例 4.3.1 是同一个。由于 $b=v=7$，所以是一个 SBIBD，其中 $b=v=7, k=r=3, \lambda=1$。

例 4.3.4 同上例，现在设初始值区组为：$B=\{0,1,4\}$。于是

$$B+0=\{0,1,4\},\quad B+1=\{1,2,5\},\quad B+2=\{2,3,6\},$$
$$B+3=\{3,4,0\},\quad B+4=\{4,5,1\},\quad B+5=\{5,6,2\},$$
$$B+6=\{6,0,3\}$$

这不是一个 BIBD，因为样品 1 和 2 同时出现在一个区组中，而样品 1 和 5 却同时出现在两个区组中。

从这两个例子可以看出，从初始值区组发展起来的区组是否是一个 SBIBD，关键是初始值区组的选取。

令 B 是 Z_v 的 k 元子集，设 $S=\{x-y \mid x\neq y, x,y\in B\}$。容易看出：$S$ 是一个重集，且含有 $k(k-1)$ 个元素（两种顺序），S 中的元素称为**差分**。如果 Z_v 中的每个非零整数都在 S 中，并且都出现 λ 次，则称 B 为 mod v **差分集**。由于 Z_v 中存在 $v-1$ 个非零整数，因此有：$\lambda(v-1)=k(k-1)$，即

$$\lambda = \frac{k(k-1)}{v-1}$$

例 4.3.5 令 $v=7, k=3$，考虑 $B=\{0,1,3\}$。计算 B 中整数的减法表

−	0	1	3
0	0	6	4
1	1	0	5
3	3	2	0

忽略对角线上的那些 0。考察该表发现，Z_7 中的每个非零整数 1，2，3，4，5，6 在非对角线位置上恰好出现一次，因此，B 是 mod 7 的一个差分集。

例 4.3.6　令 $v=7$，$k=3$，考虑 $B=\{0,1,4\}$。计算 B 中整数的减法表

−	0	1	4
0	0	6	3
1	1	0	4
4	4	3	0

可以看到，作为差分 1 和 6 每个在非对角线位置上出现一次，3 和 4 出现两次，而 2 和 5 根本不出现。因此，B 不是 mod 7 的一个差分集。

定理 4.3.3　令 B 为 Z_v 中的 $k(k<v)$ 元子集，它形成 mod v 的差分集。则从 B 作为初始值区组发展起来的区组形成一个 SBIBD，其指数

$$\lambda=\frac{k(k-1)}{v-1}$$

证明：由于 $k<v$，故由 B 这些发展起来的区组是不完全区组，并且每个区组包含 k 个元素，区组的个数 b 与样品的个数 v 相同。现在只需要证明 Z_v 的每一对元素同时属于相同个数的区组。因为 B 是差分集，所以 Z_v 的每一个非零整数作为差分恰好出现 $\lambda=\dfrac{k(k-1)}{v-1}$ 次。剩下的只需要证明 Z_v 的每一对不同的整数恰好在 λ 个区组中。

令 p 和 q 是 Z_v 中的互异整数，则 $p-q\neq 0$。由于 B 是 mod v 的差分集，从而方程

$$x-y=p-q$$

在 B 中有 λ 个解。对每个这样的解 x 和 y，令 $j=p-x$，则

$$p=x+j,\quad q=y-x+p=y+j$$

即 p 和 q 都属于 $B+j$。而这样的 j 有 λ 个，因此，p 和 q 同属于 λ 个区组。又因为

$$v(v-1)\lambda=v(v-1)\frac{k(k-1)}{v-1}=vk(k-1)$$

所以，Z_v 的每一对不同的整数恰好在 λ 个区组中。

例 4.3.7　求 Z_{11} 中大小为 5 的差分集，并用它作为初始值区组构造一个 SBIBD。

解：设 $B=\{0,2,3,4,8\}$，计算 B 的减法表

−	0	2	3	4	8
0	0	9	8	7	3
2	2	0	10	9	5
3	3	1	0	10	6
4	4	2	1	0	7
8	8	6	5	4	0

观察对角线以外的元素可以看到，Z_{11} 中的每个非零整数作为差分恰好出现两次，从而 B 是一个差分集。将 B 作为一个初始值区组按前面讨论的方法得到具有参数 $b=v=11,k=r=5,\lambda=2$ 的 SBIBD 的下面的区组：

$B+0=\{0,2,3,4,8\}$，　$B+1=\{1,3,4,5,9\}$，　$B+2=\{2,4,5,6,10\}$，
$B+3=\{0,3,5,6,7\}$，　$B+4=\{1,4,6,7,8\}$，　$B+5=\{2,5,7,8,9\}$，
$B+6=\{3,6,8,9,10\}$，　$B+7=\{0,4,7,9,10\}$，　$B+8=\{0,1,5,8,10\}$，
$B+9=\{0,1,2,6,9\}$，　$B+10=\{1,2,3,7,10\}$

4.3.2 Steiner 三元系统

设 $B=(b,v,r,k,\lambda)$是一个 BIBD，由于 B 是非完全的，所以 $k<v$。特别地，如果 $k=2,\lambda=1$，那么，B 中每个区组恰好包含两个样品，且每对样品正好在一个区组中出现。也就是说，在这种情况下区组的数量就是它的样品集合的 2 元子集的数量。例如，一个 $v=3,k=2,\lambda=1$ 的 BIBD 中的区组是下面的子集：$\{0,1\},\{0,2\},\{1,2\}$。如果取 $\lambda=2$，只要将上面的每个区组取两次。如果取 $\lambda=3$，只要将上面的每个区组取 3 次。因此，$k=2$ 的 BIBD 是平凡的。本小节主要研究 BIBD 的另一种特殊情况，即 $k=3,\lambda=1$ 的情况。

定义 4.3.2　$k=3,\lambda=1$ 的平衡区组设计叫做 **Steiner 三元系统**。

例 4.3.1 就是一个 Steiner 三元系统，并且是一个 SBIBD。这是形成 SBIBD 的 Steiner 三元系统的仅有的例子。Steiner 三元系统的另外一个例子是取 $v=3$，3 个样品 0，1 和 2 以及一个区组$\{0,1,2\}$而得到的。这时 $b=1$，显然每对样品含于 $\lambda=1$ 个区组中，因为 $v=k=3$，所以它不是一个 BIBD。每个其他的 Steiner 三元系统都是一个 BIBD。

下面的定理给出 Steiner 三元系统存在的条件。

定理 4.3.4　含有 $v(v\geqslant 2)$个样品的 Steiner 三元系统存在的充分必要条件是：存在非负整数 n，使得 $v=6n+1(n\geqslant 1)$或 $v=6n+3(n\geqslant 0)$。

证明：必要性。由式(4.4)知，在 Steiner 三元系统中

$$2r=v-1,\quad 即\quad r=\frac{v-1}{2}$$

这说明 $v-1$ 是偶数，v 是奇数。又由式(4.6)知

$$3b=\frac{v(v-1)}{2},\quad 即\quad b=\frac{v(v-1)}{6}$$

此式说明 $v(v-1)$是 6 的倍数，即 v 或 $v-1$ 能被 3 整除。如果 v 能被 3 整除，而 v 是

奇数，所以 v 是 3 乘以一个奇数，即

$$v = 3\times(2n+1) = 6n+3,\quad n\geqslant 0$$

如果 $v-1$ 能被 3 整除，又 $v-1$ 是偶数，所以 $v-1$ 是 3 乘以一个偶数，即

$$v-1 = 3\times 2n = 6n,\quad v = 3n+1$$

因为 $v\geqslant 2$，所以上式中必须 $n\geqslant 1$。必要性证毕。

对于充分性的证明，已超出了本书的范围。有兴趣的读者可以参见文献[8]。

下面的定理证明了一类特殊的 Steiner 三元系统的存在性，并给出了具体的构造方法。

定理 4.3.5　如果存在含有 v_1 个样品和 v_2 个样品的两个 Steiner 三元系统，那么就一定存在含有 v_1v_2 个样品的 Steiner 三元系统。

证明：设 B_1 是具有 v_1 个样品 $a_1,a_2,\cdots,a_{v_1}$ 的 Steiner 三元系统，B_2 是具有 v_2 个样品 $b_1,b_2,\cdots,b_{v_2}$ 的 Steiner 三元系统。设 S 是由 v_1v_2 个样品 c_{ij} 组成的集合，其中 $i=1,2,\cdots,v_1$，$j=1,2,\cdots,v_2$。把这些样品看成 v_1 行 v_2 列的矩阵的元素，矩阵的行对应 $a_1,a_2,\cdots,a_{v_1}$，矩阵的列对应 $b_1,b_2,\cdots,b_{v_2}$，如下所示：

$$\begin{array}{c} \\ a_1 \\ a_2 \\ \vdots \\ a_{v_1} \end{array}\begin{array}{c} \begin{array}{cccc} b_1 & b_2 & \cdots & b_{v_2} \end{array} \\ \begin{pmatrix} c_{11} & c_{12} & \cdots & c_{1v_2} \\ c_{21} & c_{22} & \cdots & c_{2v_2} \\ \vdots & \vdots & \ddots & \vdots \\ c_{v_11} & c_{v_12} & \cdots & c_{v_1v_2} \end{pmatrix} \end{array} \tag{4.9}$$

现在来定义 S 的元素的三元组集合 B。设 $\{c_{ir},c_{js},c_{kt}\}$ 是 S 的 3 个元素组成的集合，则 $\{c_{ir},c_{js},c_{kt}\}$ 是 B 中的一个三元组当且仅当下述命题之一成立。

(1) $r=s=t$，且 $\{a_i,a_j,a_k\}$ 是 B_1 的三元组。换句话说，c_{ir}、c_{js} 和 c_{kt} 在矩阵(4.9)的同一列上，且它们所在的行对应 B_1 的一个三元组。

(2) $i=j=k$，且 $\{b_r,b_s,b_t\}$ 是 B_2 的三元组。换句话说，c_{ir}、c_{js} 和 c_{kt} 在矩阵(4.9)的同一行上，且它们所在的列对应 B_2 的一个三元组。

(3) i,j,k 互异且 $\{a_i,a_j,a_k\}$ 是 B_1 的三元组，而 r,s,t 互异且 $\{b_r,b_s,b_t\}$ 是 B_2 的三元组。换句话说，c_{ir}、c_{js} 和 c_{kt} 在矩阵(4.9)的不同行不同列上，而它们所在的行对应 B_1 的一个三元组，它们所在的列对应 B_2 的一个三元组。

由 B 的定义可知如下事实：没有 B 的三元组恰好位于矩阵(4.9)的两行或者恰好位于矩阵(4.9)的两列。以下的证明基于这个事实。

下面证明如上定义的 S 的三元组集合 B 构成一个 Steiner 三元系统。

令 c_{ir}、c_{js} 为 S 的一对不同的元素，现在需要证明存在唯一的既含 c_{ir} 又含 c_{js} 的 B 的三元组。分 3 种情况讨论。

情形 1：$r=s$，因而 $i\neq j$。此时元素对是位于矩阵(4.9)的同一列上的元素 c_{ir}、c_{jr}。由于 B_1 是一个 Steiner 三元系统，因此存在 B_1 的包含互异对 a_i、a_j 的唯一的三元组 $\{a_i,a_j,a_k\}$。从而 $\{c_{ir},c_{jr},c_{kr}\}$ 是 B 的唯一包含 c_{ir}、c_{jr} 的唯一的三元组。

情形 2：$i=j$，因而 $r\neq s$。此时元素对是位于矩阵(4.9)的同一行上的元素 c_{ir}、

c_{is}。由于 B_2 是一个 Steiner 三元系统，因此存在 B_2 的包含互异对 b_r、b_s 的唯一的三元组$\{b_r,b_s,b_t\}$。从而$\{c_{ir},c_{is},c_{it}\}$是 B 的包含 c_{ir}、c_{is} 的唯一的三元组。

情形 3：$i\neq j$ 且 $r\neq s$。存在 B_1 的包含互异对 a_i、a_j 的唯一的三元组$\{a_i,a_j,a_k\}$和 B_2 的包含互异对 b_r、b_s 的唯一的三元组$\{b_r,b_s,b_t\}$。因此$\{c_{ir},c_{js},c_{kt}\}$是 B 的包含 c_{ir}、c_{js} 的唯一的三元组。

这样就证明了 B 是具有 v_1v_2 个样品的 Steiner 三元系统。

例 4.3.8 令 B_1 是具有 3 个样品 a_1,a_2,a_3 和唯一一个三元组$\{a_1,a_2,a_3\}$的 Steiner 三元系统，B_2 是具有 3 个样品 b_1,b_2,b_3 和唯一一个三元组$\{b_1,b_2,b_3\}$的 Steiner 三元系统。现在利用定理 4.3.5 的证明中给出的构造方法构造一个具有 $3\times3=9$ 个样品的 Steiner 三元系统 B。设 S 是 9 个样品的集合，它的元素是下列矩阵的元素：

$$\begin{array}{c} \\ a_1 \\ a_2 \\ a_3 \end{array}\begin{array}{c} \begin{array}{ccc} b_1 & b_2 & b_3 \end{array} \\ \begin{pmatrix} c_{11} & c_{12} & c_{13} \\ c_{21} & c_{22} & c_{23} \\ c_{31} & c_{32} & c_{33} \end{pmatrix} \end{array}$$

由定理 4.3.5 的证明中的构造方法得到具有 9 个样品和 12 个三元组的 Steiner 三元系统 B。

（1）3 行中每行的元素如下：

$$\{c_{11},c_{12},c_{13}\},\quad \{c_{21},c_{22},c_{23}\},\quad \{c_{31},c_{32},c_{33}\}$$

（2）3 列中每列的元素如下：

$$\{c_{11},c_{21},c_{31}\},\quad \{c_{12},c_{22},c_{32}\},\quad \{c_{13},c_{23},c_{33}\}$$

（3）在不同行不同列的元素如下：

$$\{c_{11},c_{22},c_{33}\},\quad \{c_{12},c_{23},c_{31}\},\quad \{c_{13},c_{21},c_{32}\}$$
$$\{c_{13},c_{22},c_{31}\},\quad \{c_{12},c_{21},c_{33}\},\quad \{c_{11},c_{23},c_{32}\}$$

如果用 0,1,2,3,4,5,6,7,8 分别代替 $c_{11},c_{21},c_{31},c_{12},c_{22},c_{32},c_{13},c_{23},c_{33}$，则 Steiner 三元系统 B 可以写成下面简单的形式：

$$\begin{array}{cccc} \{0,1,2\} & \{0,3,6\} & \{0,4,8\} & \{2,4,6\} \\ \{3,4,5\} & \{1,4,7\} & \{2,3,7\} & \{1,3,8\} \\ \{6,7,8\} & \{2,5,8\} & \{1,5,6\} & \{0,5,7\} \end{array} \tag{4.10}$$

式(4.10)的列将 B 的 12 个三元组分成了 4 部分，使每一个样品恰好出现在每一部分的一个三元组中，即每一部分都是 B 的样品集分成三元组的一个划分。具有这个性质的 Steiner 三元系统称为**可解的**(resolvable)，而每一部分叫做**可解类**。

Steiner 三元系统的可解性的概念首先产生于下面的由 Kirkman 提出的问题，称为 Kirkman 女生问题。

学校女教师带领她班上 15 个女生进行日常操练。这些女生被排成 5 行，每行 3 人，因此每个女孩有两个队友。问能否计划连续操练 7 天，使得没有一个女孩与她的任何同学在 3 人行中操练的次数超过一次？

这个问题的解由 15 个女孩的 $7\times5=35$ 个三元组组成，要求：每一对女孩恰好

同在一个三元组中，且能够把35个三元组划分成7类，每类5个三元组，使得每类中每个女孩恰好出现在一个三元组中，即每类都是这15个女孩的一个三元组划分。由定理4.3.4的证明知，具有$v=15$个样品的Steiner三元系统的三元组的个数为

$$b=\frac{v(v-1)}{6}=35$$

所以，Kirkman女生问题就是求解一个具有$v=15$个样品的Steiner三元系统。这样的Steiner三元系统与将它们分成的7个部分(每一部分对应7天中的一天)表示如下：

{0,1,2}	{0,3,4}	{0,5,6}	{0,7,8}	{0,9,10}	{0,11,12}	{0,13,14}
{3,7,11}	{1,7,9}	{1,8,10}	{1,11,13}	{1,12,14}	{1,3,5}	{1,4,6}
{4,9,14}	{2,12,13}	{2,11,14}	{2,4,5}	{2,3,6}	{2,8,9}	{2,7,10}
{5,10,12}	{5,8,14}	{3,9,13}	{3,10,14}	{4,8,11}	{4,10,13}	{3,8,12}
{6,8,13}	{6,10,11}	{4,7,12}	{6,9,12}	{5,7,13}	{6,7,10}	{5,9,11}

例4.3.8实际上给出了9个女孩情况下Kirkman女生问题的解。在这种情况下，共有9个女生并安排她们进行4天日常操练，每个女孩在所有4天中都有不同的队友。

关于Kirkman女生问题的进一步的讨论参见文献[9]。

4.3.3 拉丁方

定义4.3.3 令n是一个正整数，S是一个n集合，$\boldsymbol{A}$是S上的一个n阶方阵。如果$\boldsymbol{A}$的每一行都是S的n个元素的一个排列，同时$\boldsymbol{A}$的每一列也都是S的n个元素的一个排列，则称$\boldsymbol{A}$是S上的一个n阶**拉丁方**。

S的元素的具体性质并不重要，因此通常把S取为$Z_n=\{0,1,2,\cdots,n-1\}$。此时将拉丁方$\boldsymbol{A}$的行列计数为$0,1,2,\cdots,n-1$。下面是几个拉丁方的例子：

$$\begin{pmatrix}0&1\\1&0\end{pmatrix},\quad \begin{pmatrix}0&1&2\\1&2&0\\2&0&1\end{pmatrix},\quad \begin{pmatrix}0&1&2&3\\1&2&3&0\\2&3&0&1\\3&0&1&2\end{pmatrix}$$

上面这几个拉丁方有一个共同的特点，就是它们的第0行是按自然顺序排列的，这种拉丁方称之为**标准型**。

考虑Z_n上的n阶拉丁方$\boldsymbol{A}$。设$k\in Z_n$，则k在$\boldsymbol{A}$中出现n次，并出现在$\boldsymbol{A}$的不同行不同列上。令$A(k)$为$k(k=0,1,2,\cdots,n-1)$在$\boldsymbol{A}$中所占据的位置的集合，则

$$A(0),A(1),\cdots,A(n-1)$$

是$\boldsymbol{A}$的n^2个位置的一个划分。注意到，在拉丁方中，如果把1和2这两个数互换，其结果仍是一个拉丁方。对应于上面的划分就相当于把$A(1)$变成$A(2)$并把$A(2)$变成$A(1)$。更一般地，任意交换$A(0),A(1),\cdots,A(n-1)$，所得结果仍然是一个拉丁方。用这种方法可以产生$n!$个拉丁方，并且容易看出通过交换$A(0),A(1),\cdots,A(n-1)$，总可以把拉丁方化成标准型。

定理 4.3.6 令 n 是一个正整数，$\boldsymbol{A}$ 是一个 n 阶方阵，并且位于 i 行 j 列上的元素 a_{ij} 为

$$a_{ij} = i + j \pmod n, \quad i,j = 0,1,2,\cdots,n-1$$

则 $\boldsymbol{A}$ 是 Z_n 上的一个 n 阶拉丁方。

证明： 考察矩阵 $\boldsymbol{A}$ 的第 $i(i=0,1,2,\cdots,n-1)$ 行。假设 $a_{ij}=a_{ik}$，即

$$i + j = i + k$$

两边加上 i 在 Z_n 中的加法负元 $-i$，得 $j=k$，这就证明了在 $\boldsymbol{A}$ 的任一行上不存在相同的元素。同理可证，在 $\boldsymbol{A}$ 的任一列上也不存在相同的元素，即 $\boldsymbol{A}$ 是 Z_n 上的一个 n 阶拉丁方。

定理 4.3.6 中构造的 n 阶拉丁方只不过是 Z_n 中的加法表。下面给出构造拉丁方的更一般的方法。

定理 4.3.7 令 n 是一个正整数，r 是 Z_n 中的非零元，且 $(r,n)=1$。又令 $\boldsymbol{A}$ 是一个 n 阶方阵，并且位于 i 行 j 列上的元素 a_{ij} 为

$$a_{ij} = r \times i + j \pmod n, \quad i,j = 0,1,2,\cdots,n-1$$

则 $\boldsymbol{A}$ 是 Z_n 上的一个 n 阶拉丁方。

证明： 同定理 4.3.6 的证明类似，可以证明在 $\boldsymbol{A}$ 的任一行上不存在相同的元素。下证在 $\boldsymbol{A}$ 的任一列上也不存在相同的元素。因为 $(r,n)=1$，则 r 在 Z_n 中有乘法逆元 r^{-1}。考察矩阵 $\boldsymbol{A}$ 的第 $j(j=0,1,2,\cdots,n-1)$ 列。假设 $a_{ij}=a_{kj}$，即 $r\times i+j=r\times k+j$，亦即

$$r \times (i-k) = 0$$

两边乘以 r^{-1} 得 $i=k$，这就证明了在 $\boldsymbol{A}$ 的任一列上也不存在相同的元素。因此 $\boldsymbol{A}$ 是 Z_n 上的一个 n 阶拉丁方。

显然定理 4.3.6 就是定理 4.3.7 在 $r=1$ 时的特殊情况。

定理 4.3.7 中用 Z_n 的乘法可逆元 r 构造的拉丁方记为 L_n^r。容易证明：如果 r 不是 Z_n 的乘法可逆元，则用定理 4.3.7 中的方法构造的矩阵一定不是拉丁方。

定义 4.3.4 设 $\boldsymbol{A}=(a_{ij})$ 和 $\boldsymbol{B}=(b_{ij})$ 是 Z_n 上的两个 n 阶拉丁方。如果 n^2 个有序对 (a_{ij},b_{ij}) 互不相同，则称 $\boldsymbol{A}$、$\boldsymbol{B}$ 是**正交拉丁方**。

例 4.3.9 不存在两个 2 阶的正交拉丁方，因为 2 阶拉丁方只有两个：

$$\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

显然它们不是正交的。考察下面两对拉丁方：

$$\begin{pmatrix} 0 & 1 & 2 \\ 1 & 2 & 0 \\ 2 & 0 & 1 \end{pmatrix}, \begin{pmatrix} 0 & 1 & 2 \\ 2 & 0 & 1 \\ 1 & 2 & 0 \end{pmatrix} \rightarrow \begin{pmatrix} (0,0) & (1,1) & (2,2) \\ (1,2) & (2,0) & (0,1) \\ (2,1) & (0,2) & (1,0) \end{pmatrix}$$

$$\begin{pmatrix} 0 & 1 & 2 & 3 \\ 1 & 0 & 3 & 2 \\ 2 & 3 & 0 & 1 \\ 3 & 2 & 1 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 1 & 2 & 3 \\ 3 & 2 & 1 & 0 \\ 1 & 0 & 3 & 2 \\ 2 & 3 & 0 & 1 \end{pmatrix} \rightarrow \begin{pmatrix} (0,0) & (1,1) & (2,2) & (3,3) \\ (1,3) & (0,2) & (3,1) & (2,0) \\ (2,1) & (3,0) & (0,3) & (1,3) \\ (3,2) & (2,3) & (1,0) & (0,1) \end{pmatrix}$$

每一对拉丁方相应位置的元素组成的有序对恰好出现一次，所以是正交的。

正交拉丁方的概念源于 Euler 提出的著名的 36 名军官的问题，其表述如下。

有来自 6 个团队且分属 6 种军衔的 36 名军官，每个团 6 名，每个军衔 6 名。问能否把这 36 名军官排成 6 行 6 列的方队，使得每行与每列的 6 名军官既有不同的军衔又来自不同的团队？

用 0,1,2,3,4,5 把军衔和团队编号，这样每个军官可用 0,1,2,3,4,5 所构成的二元有序对表示，其第一个分量表示该军官的军衔，第二个分量表示该军官所属的团队。因此 36 名军官的问题就归结为能否构造一对 6 阶正交拉丁方。Euler 在 1782 年猜想这样的对不存在，直到 1900 年左右，Tarry 通过枚举法证明 Euler 猜想是正确的，即不存在一对 6 阶的正交拉丁方。

现在把正交拉丁方的概念推广到任意有限个拉丁方上去。

定义 4.3.5　设 $\boldsymbol{A}_1,\boldsymbol{A}_2,\cdots,\boldsymbol{A}_k$ 是 Z_n 上的一族 n 阶拉丁方。如果它们中的任意两个都是正交的，则称 $\boldsymbol{A}_1,\boldsymbol{A}_2,\cdots,\boldsymbol{A}_k$ 是正交的，并把它们称为**正交拉丁方族**。

定理 4.3.8　设 $\boldsymbol{A}_1,\boldsymbol{A}_2,\cdots,\boldsymbol{A}_k$ 是 Z_n 上的 k 个 n 阶正交拉丁方，则 $k\leqslant n-1$。

证明：由前面的讨论可知，每一个拉丁方 $\boldsymbol{A}_1,\boldsymbol{A}_2,\cdots,\boldsymbol{A}_k$ 都可以化为标准型，并且容易证明这并不影响它们的相互正交性。因此可以假设这 k 个拉丁方 $\boldsymbol{A}_1,\boldsymbol{A}_2,\cdots,\boldsymbol{A}_k$ 都是标准型，即它们第 0 行位置的元素都是按自然顺序 $0,1,2,\cdots,n-1$ 排列的。现在考察这 k 个拉丁方在第 1 行第 0 列位置上的元素。由正交性可知，这 k 个元素互不相同，同时它们都不等于 0，并且都是 Z_n 上的元素。所以 $k\leqslant n-1$。

如果定理 4.3.8 中的等号成立，即 $k=n-1$，则称这个正交族是**完备的**。

下面的定理说明，当 n 是素数时，可以构造出 $n-1$ 个 n 阶的正交拉丁方。

定理 4.3.9　若 n 是素数，则 $\boldsymbol{L}_n^1,\boldsymbol{L}_n^2,\cdots,\boldsymbol{L}_n^{n-1}$ 是 $n-1$ 个 n 阶的正交拉丁方。

证明：因 n 是素数，则 Z_n 的非零元都是可逆的。由定理 4.3.7，$\boldsymbol{L}_n^1,\boldsymbol{L}_n^2,\cdots,\boldsymbol{L}_n^{n-1}$ 都是 n 阶拉丁方。令 r 和 s 为 Z_n 中任意两个互异的非零整数，下证 $\boldsymbol{L}_n^r$ 和 $\boldsymbol{L}_n^s$ 是正交的。假设 $\boldsymbol{L}_n^r$ 和 $\boldsymbol{L}_n^s$ 相应位置元素构成的 n^2 个有序对中有两个是相同的。不妨设 i 行 j 列上的有序对和 k 行 l 列上的有序对相同，其中 $i\neq k$ 或者 $j\neq l$。由定理 4.3.7 中 $\boldsymbol{L}_n^r$ 和 $\boldsymbol{L}_n^s$ 的定义知

$$r\times i+j=r\times k+l \quad 及 \quad s\times i+j=s\times k+l$$

改写这个方程得到

$$r\times(i-k)=l-j \quad 及 \quad s\times(i-k)=l-j$$

从而

$$r\times(i-k)=s\times(i-k)$$

若 $i\neq k$，即 $i-k\neq 0$，从而它是 Z_n 中的可逆元，上式等式两边同时乘以 $i-k$ 的逆元 $(i-k)^{-1}$，得 $r=s$，与 r 和 s 互异矛盾，因此必须使 $i=k$。再将 $i=k$ 代入前一个方程，得 $j=l$，与假设矛盾。所以 $\boldsymbol{L}_n^r$ 和 $\boldsymbol{L}_n^s$ 相应位置元素构成的 n^2 个有序对中没有两个是相同的，也就是说 $\boldsymbol{L}_n^r$ 和 $\boldsymbol{L}_n^s$ 是正交的。由 r 和 s 的任意性知，$\boldsymbol{L}_n^1,\boldsymbol{L}_n^2,\cdots,\boldsymbol{L}_n^{n-1}$ 是正交的拉丁方族。

我们知道，有限域中元素的个数总是一个素数的幂。反过来，对于每一个素数 p

和每一个正整数 k 总存在含有 p^k 个元素的有限域。现在把定理 4.3.7 和定理 4.3.9 推广到有限域上。

设 F 是含有 $n=p^k$ 个元素的有限域，其中 p 是素数，k 是正整数。记 F 的元素为

$$a_0 = 0, a_1, a_2, \cdots, a_{n-1}$$

a_0 是 F 的零元。考虑 F 的任意非零元 $a_r(r\neq 0)$，并定义 n 阶方阵 $\boldsymbol{A}=(a_{ij})$ 如下：

$$a_{ij} = a_r \times a_i + a_j \quad i,j = 0,1,2,\cdots,n-1$$

其中的运算是域 F 中的运算。完全类似于定理 4.3.7 的证明，知 $\boldsymbol{A}$ 是 F 上的 n 阶拉丁方。用 $\boldsymbol{L}_n^{a_r}$ 表示以这种方法构造的拉丁方，于是按照定理 4.3.9 的证明，得到 $n-1$ 个 n 阶的正交拉丁方族：

$$\boldsymbol{L}_n^{a_1}, \boldsymbol{L}_n^{a_2}, \cdots, \boldsymbol{L}_n^{a_{n-1}}。 \tag{4.11}$$

把上面的事实概括为下面的定理。

定理 4.3.10 令 $n=p^k$ 是一个整数，它是素数 p 的幂。则存在 $n-1$ 个 n 阶的正交拉丁方族。事实上，从含有 $n=p^k$ 个元素的有限域构造的 $n-1$ 个 n 阶拉丁方(4.11)就是 $n-1$ 个 n 阶的正交拉丁方。

例 4.3.10 显然 x^2+x+1 是 Z_2 上的不可约多项式。设 α 是 x^2+x+1 的一个根，把 α 添加到 Z_2 上得到一个四元域 F。因为 α 是 x^2+x+1 的一个根，所以 $\alpha^2+\alpha+1=0$，即

$$\alpha^2 = -\alpha - 1 = \alpha + 1$$

因此，$F=\{0,1,\alpha,\alpha^2\}$。在此四元域上，应用定理 4.3.9 的构造方法得到下列拉丁方：

$$\boldsymbol{L}_4^1 = \begin{pmatrix} 0 & 1 & \alpha & \alpha^2 \\ 1 & 0 & \alpha^2 & \alpha \\ \alpha & \alpha^2 & 0 & 1 \\ \alpha^2 & \alpha & 1 & 0 \end{pmatrix}, \quad \boldsymbol{L}_4^{\alpha} = \begin{pmatrix} 0 & 1 & \alpha & \alpha^2 \\ \alpha & \alpha^2 & 0 & 1 \\ \alpha^2 & \alpha & 1 & 0 \\ 1 & 0 & \alpha^2 & \alpha \end{pmatrix}, \quad \boldsymbol{L}_4^{\alpha^2} = \begin{pmatrix} 0 & 1 & \alpha & \alpha^2 \\ \alpha^2 & \alpha & 1 & 0 \\ 1 & 0 & \alpha^2 & \alpha \\ \alpha & \alpha^2 & 0 & 1 \end{pmatrix}$$

直接验证可知，$\boldsymbol{L}_4^1, \boldsymbol{L}_4^{\alpha}, \boldsymbol{L}_4^{\alpha^2}$ 是 F 上的 3 个 4 阶正交拉丁方。

下面的定理给出了 $n-1$ 个 n 阶的正交拉丁方与区组设计之间的一种关系。

定理 4.3.11 令 $n\geqslant 2$ 为一整数。如果存在 $n-1$ 个 n 阶的正交拉丁方，则存在具有参数

$$b = n^2 + n, \quad v = n^2, \quad k = n, \quad r = n+1, \quad \lambda = 1 \tag{4.12}$$

的可解的 BIBD。反之，如果存在具有式(4.12)中参数的 BIBD，则存在 $n-1$ 个 n 阶的正交拉丁方。

证明： 先证定理的前半部分。设 $\boldsymbol{A}_1, \boldsymbol{A}_2, \cdots, \boldsymbol{A}_{n-1}$ 是 $n-1$ 个 n 阶的正交拉丁方，下面使用 $n+1$ 个矩阵

$$\boldsymbol{R}_n, \boldsymbol{S}_n, \boldsymbol{A}_1, \boldsymbol{A}_2, \cdots, \boldsymbol{A}_{n-1} \tag{4.13}$$

来构造具有式(4.12)中参数的可解 BIBD，其中

$$\boldsymbol{R}_n = \begin{pmatrix} 0 & 0 & \cdots & 0 \\ 1 & 1 & \cdots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ n-1 & n-1 & \cdots & n-1 \end{pmatrix}, \quad \boldsymbol{S}_n = \begin{pmatrix} 0 & 1 & \cdots & n-1 \\ 0 & 1 & \cdots & n-1 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 1 & \cdots & n-1 \end{pmatrix}$$

令 $A_i(k)$表示 $\boldsymbol{A}_i$ 被 $k(k=0,1,\cdots,n-1)$占据的位置的集合。因为 $\boldsymbol{A}_i$ 是一拉丁方，所以 $A_i(k)$包含每一行每一列上的位置，且 $A_i(k)$没有属于同一行或属于同一列的两个位置。对 $\boldsymbol{R}_n$ 和 $\boldsymbol{S}_n$ 也用这个记号。

将样品集 V 取作 $n\times n$ 矩阵的 $v=n^2$ 个位置的集合，即

$$V=\{(i,j)\mid i,j=0,1,\cdots,n-1\}$$

式(4.13)的 $n+1$ 个矩阵的每一个都确定 n 个区组：

$$R_n(0)\quad R_n(1)\quad \cdots\quad R_n(n-1) \tag{4.14}$$

$$S_n(0)\quad S_n(1)\quad \cdots\quad S_n(n-1) \tag{4.15}$$

$$\begin{matrix} A_1(0) & A_1(1) & \cdots & A_1(n-1) \\ \vdots & \vdots & & \vdots \\ A_{n-1}(0) & A_{n-1}(1) & \cdots & A_{n-1}(n-1) \end{matrix} \tag{4.16}$$

这样就有 $b=n\times(n+1)=n+n^2$ 个区组，每个区组含有 $k=n$ 个样品。令 B 表示这些区组的集合。为了证明 B 是具有参数(4.12)的 BIBD，只需验证每一对样品恰好在 $\lambda=1$ 个区组中。有下面 3 种可能情况。

(1) 同行的两个样品：它们恰好出现在式(4.14)的一个区组中，但不在其他区组中。

(2) 同列的两个样品：它们恰好出现在式(4.15)的一个区组中，但不在其他区组中。

(3) 两个样品(i,j)和(p,q)属于不同的行和不同的列：显然，这两个样品不会同时存在于式(4.14)和式(4.15)的任意一个区组中。假设它们同时存在于区组 $A_r(e)$和 $A_s(f)$中，这意味着：$\boldsymbol{A}_r$ 的 i 行 j 列的位置和 p 行 q 列的位置存在着元素 e，且在 $\boldsymbol{A}_s$ 的同样的位置存在着元素 f。如果 $r\neq s$，那么有序对(e,f)出现两次，与 $\boldsymbol{A}_r$ 和 $\boldsymbol{A}_s$ 的正交性矛盾。所以 $r=s$，这说明 $\boldsymbol{A}_r$ 同时有 e 和 f 在 i 行 j 列的位置和 p 行 q 列的位置，于是必有 $e=f$。因此 $A_r(e)$和 $A_s(f)$是同一区组。由此可以断定，(i,j)和(p,q)至多在同一个区组中。

下面来说明每一对样品恰好在一个区组中。已知共有 n^2 个样品，它们可以构成$\frac{n^2(n^2-1)}{2}$个对。每一对至多在一个区组中，共有 $n+n^2$ 个区组，每个区组有 n 个样品，从而包含$\frac{n(n-1)}{2}$个对，于是对于所有的区组共有

$$(n^2+n)\times\frac{n(n-1)}{2}=\frac{n^2(n^2-1)}{2}$$

个对，而这恰好又是样品的总对数。因此由鸽巢原理，每一对样品恰好在一个区组中。从而 B 是具有参数(4.12)的 BIBD。

容易看出上面所构造的 B 是可解的。n^2+n 个区组的集合分成 $n+1$ 部分(可解类)，每部分 n 个区组(见式(4.14)、式(4.15)和式(4.16))，而每个可解类均为 n^2 个样品的一个划分。

对于定理后半部分的证明，仅叙述构造 $n-1$ 个 n 阶的正交拉丁方的方法，具体

细节留给读者自己去验证。设有一个具有参数(4.12)的可解的 BIBD,记为 B。因为有n^2 个样品,并且每个区组含有 n 个样品,所以每个可解类包含 n 个区组。又由于存在 $n+n^2$ 个区组,因此存在 $n+1$ 个可解类

$$B_1, B_2, \cdots, B_{n+1}$$

令 B_n 中的区组是

$$H_0, H_1, \cdots, H_{n-1}$$

并令 B_{n+1} 中的区组是

$$V_0, V_1, \cdots, V_{n-1}$$

所以任意取定一个样品 x,一定存在 $H_0, H_1, \cdots, H_{n-1}$ 中唯一的一个区组 H_i,使 $x\in H_i$,同样也存在 $V_0, V_1, \cdots, V_{n-1}$ 中唯一的一个区组 V_j,使 $x\in V_j$。这样每一个样品 x 都对应于一个有序对(i,j),又因为 $\lambda=1$,所以两个不同的样品不会与相同的有序对对应。于是可以将样品集看成是有序对组成的集合:

$$V=\{(i,j)\mid i,j=0,1,\cdots,n-1\}$$

现在考虑其他的可解类 $B_p(p=1,2,\cdots,n-1)$。将 B_p 中的区组记为

$$A_p(0), A_p(1), \cdots, A_p(n-1)$$

这些区组将 V 划分成 n 个大小为 n 的集合。做 $n\times n$ 矩阵 $\boldsymbol{A}_p$,使它在位置集合 $A_p(k)$的每个位置上的元素均为 k。显然 $\boldsymbol{A}_p$ 是一个拉丁方,事实上,如果在 $\boldsymbol{A}_p$ 的第 i 行有两个 k,那么就存在两个样品(i,a)和(i,b),它们既在区组 H_i 中,又在区组 $A_i(k)$中。并且对于 $p\neq q$,可证 $\boldsymbol{A}_p$ 和 $\boldsymbol{A}_q$ 是正交的(留给读者自己验证)。因此 $\boldsymbol{A}_1$, $\boldsymbol{A}_2, \cdots, \boldsymbol{A}_{n-1}$是 $n-1$ 个 n 阶的正交拉丁方。

定理 4.3.12 设 $n\geqslant 3, k\geqslant 2$ 为整数,则存在一个由 $k-2$ 个 n 阶拉丁方组成的正交族等价于存在一个 $n^2\times k$ 矩阵

$$\boldsymbol{A}=(a_{ij}) \quad i=1,2,\cdots,n^2; j=1,2,\cdots,k$$

其中 $\boldsymbol{A}$ 的元素是 $1,2,\cdots,n$,而且 $\boldsymbol{A}$ 的每一个 $n^2\times 2$ 子阵列的 n^2 行表示出了 $1,2,\cdots,n$ 的全部 n^2 个重复 2 组合。

证明:设有满足上述条件的矩阵 $\boldsymbol{A}$。对 $\boldsymbol{A}$ 做行置换得到矩阵 $\boldsymbol{B}$,使 $\boldsymbol{B}$ 的前两列所成的 $n^2\times 2$ 子阵列的行元素排成自然顺序

$$(1,1),(1,2),\cdots,(1,n),(2,1),(2,2),\cdots,(2,n),\cdots,(n,1),(n,2),\cdots,(n,n)$$

然后把 $\boldsymbol{B}$ 的第 $r(r=3,4,\cdots,k)$列上的 n^2 个元素按下面的方法构成一个 $n\times n$ 矩阵 $\boldsymbol{B}_r$:$\boldsymbol{B}_r$ 的第一行由 $\boldsymbol{B}$ 的第 r 列的前 n 个元素组成,$\boldsymbol{B}_r$ 的第二行由 $\boldsymbol{B}$ 的第 r 列的第 $n+1$ 到第 $n+n$ 这 n 个元素组成,……,如此一直到 $\boldsymbol{B}_r$ 的第 n 行由 $\boldsymbol{B}$ 的第 r 列的最后 n 个元素组成。容易证明,如此作出的 $\boldsymbol{B}_3, \boldsymbol{B}_4, \cdots, \boldsymbol{B}_k$ 一定是正交拉丁方族。事实上,由 $\boldsymbol{B}$ 的第一行性质可知 $\boldsymbol{B}_r$ 的每一行上没有相同元素,由 $\boldsymbol{B}$ 的第一列性质可知 $\boldsymbol{B}_r$ 的每一列上没有相同元素,所以 $\boldsymbol{B}_r$ 是拉丁方。另外,如果 $r\neq s$,则由 $\boldsymbol{B}$ 的 r,s 两列构成的 $n^2\times 2$ 子阵列的性质知,$\boldsymbol{B}_r$ 和 $\boldsymbol{B}_s$ 是正交的。

把上面的证明反过来就得到定理的另一半的证明。

为了方便起见,把定理 4.3.12 中定义的 $n^2\times k$ 矩阵称为**正交阵列**(orthogonal arrary),记为 $OA(k,n)$。

4.4　应用举例

4.4.1　基于正交阵列的认证码

认证码是保障信息真实性的一种信息安全技术。假设 Alice 向 Bob 发送一条信息，Alice 可以通过电子邮件或传真或移动电话把信息发送出去，而这些信道都是不安全的。Bob 想要确认这一信息是否真的由 Alice 发出，并且还要确认有没有人改动过 Alice 发出的信息。这个安全问题就可用认证码技术来解决。

下面考虑一个名叫 Oscar 的外来者攻击从 Alice 发送到 Bob 的信息的可能性。假设 Oscar 能够简单地假冒 Alice 给 Bob 发送信息，或者 Oscar 能够更改由 Alice 发送的信息。为了防止 Oscar 的攻击，Alice 在发送信息的同时发送一个认证码。设 M 是所有可能信息的集合，C 是认证码的集合，K 是密钥的集合。Alice 和 Bob 在会面时或通过一个安全的通道，事先选定一个密钥。假定他们从 K 中随机选取密钥，而与每一个密钥 $k\in K$ 相关联的是一个认证规则(authentication rule)r_k，它把一个认证码 $r_k(m)\in C$ 指定给每一条信息 $m\in M$。如果 Alice 想要把信息 m 发送给 Bob，那么她要发出信息(m,c)，其中 $c=r_k(m)$。当 Bob 接收到信息(m,c)时，他要检查 c 是否是 $r_k(m)$。如果不是，那么 Bob 有理由相信 Oscar 做了什么，而且他怀疑这条信息的真实性。当然也有这样的可能：Oscar 正确地猜到了 $r_k(m)$，这时这个过程就不能察觉到 Oscar 的所有攻击。然而，如果发生这种可能性的情况很小，而且与发送的实际信息无关，那么 Alice 和 Bob 将感到满意。

可以使用定理 4.3.12 中定义的正交阵列 $OA(p,n)$来构建认证规则。假设 $M=\{1,2,\cdots,p\}$，$C=\{1,2,\cdots,n\}$，$K=\{1,2,\cdots,n^2\}$。设矩阵 $\boldsymbol{A}$ 是一个正交阵列 $OA(p,n)$，且行由 K 的元素表示，而列由 M 的元素表示。定义 $r_k(m)=a_{km}$。

如果 Oscar 把一条信息(m,c)发送给 Bob，那么 $c=r_k(m)$的概率是多少呢？我们称这一概率为假冒概率(impersonation probability)。可以假设 Oscar 知道矩阵 $\boldsymbol{A}$，但不知道正在使用哪个密钥 $k\in K$。给定 m 和 c，那么矩阵 $\boldsymbol{A}$ 的 n^2 行中有 n 行 i 可能使得 $a_{im}=c$。因此，如果 $c=r_k(m)$，那么 Oscar 选中某行 i 使得 $r_i(m)=c$ 的概率为$\frac{n}{n^2}=\frac{1}{n}$。因此这一认证规则使得 Oscar 每 n 次只有一次机会能成功假冒 Alice。

如果 Oscar 简单地用另一条信息(m',c')替换掉 Alice 发出的信息(m,c)，情况又如何呢？即 $c'=r_k(m')$的概率是多少呢？我们称这一概率为欺骗概率(deception probability)。换句话说，欺骗概率是这样的概率，Bob 认为他接收到的信息是可信的，因此落入到 Oscar 的欺骗之中。Oscar 看到 Alice 发送的信息(m,c)，所以他知道 $r_k(m)=c$，但他不知道 k 的值。他不得不希望 $r_k(m')=c'$。对于 $\boldsymbol{A}$ 的任意两列 m 和 m'，序对(c,c')正好出现在 $\boldsymbol{A}$ 的某一行的这两列上。在 $\boldsymbol{A}$ 中存在 n 个行，在这些行中 c 出现在列 m 上。因此如果 Oscar 随机从这些行中选出一行，那么他选取的行中 c' 出现在列 m 上的概率是$\frac{1}{n}$。因此欺骗概率是$\frac{1}{n}$。

Alice 和 Bob 所选取的认证码依赖于他们希望假冒概率和欺骗概率小到什么程度。对于认证码的更多内容请参见文献[4]。

4.4.2 基于正交阵列的门限方案

在实际应用中往往存在这样的情形,某一项决策或行动非常敏感,需要一个小组里的若干成员的同意。例如,发动核攻击的密码,或银行里需要若干人同时打开的保险库等都属于这种情况。这个安全问题就可用门限方案来解决。

假设 I 是由 p 个人组成的集合,k 是启动某项行动(如打开保险库或发动核攻击)的密钥,$q\geqslant 2$ 是一个固定的整数,而且我们希望确认小组中的任意 q 个人可以一起决定 k,而只要少于 q 个人都决定不了 k。在较高的概率下实现这一功能的方法称为一个(q,p)门限方案((q,p)-threshold scheme)。固定一个密钥集 K,并确定一个不在I 中的管理者,这位管理者发给每个人关于 k 的部分信息,这些信息来自于部分信息的集合 P。这位管理者必须做到:任意 q 个人的部分信息足以计算出 k,而任意少于 q 个人的部分信息则计算不出 k。考虑 $q=2$ 的情况。

假定 $K=P=\{1,2,\cdots,n\}$,且设 $\mathbf{A}=(a_{ij})(i=1,2,\cdots,n^2;j=1,2,\cdots,p+1)$是一个 $OA(p+1,n)$。把 $\mathbf{A}$ 的前 p 列与参与者联系起来,而把最后一列与密钥联系起来。在这个小组中的所有人都知道矩阵 $\mathbf{A}$。给定 $k\in K$,设 $R_k=\{i:a_{i,p+1}=k\}$,它表示 $\mathbf{A}$ 的最后一列等于 k 的那些行组成的集合,这位管理者随机选出某行 $i\in R_k$,并把部分信息 a_{iu}给第 u 个人(把此人仍记为 u)。

第 u 人和第 v 人能决定密钥 k 吗?假设 u 得到部分信息 p_u 而 v 得到部分信息 p_v,因为 $\mathbf{A}$ 是一个 $OA(p+1,n)$,所以存在唯一一行 i,使得 $a_{iu}=p_u$,$a_{iv}=p_v$,即 u、v 能决定 i,因此能够寻找到 $a_{i,p+1}$,这就是他们所需要的密钥 k。

任何一个人 u 只基于他的部分信息 p_u 可以决定密钥 k 吗?对于密钥的任意可能值 k',存在唯一一行 i,对于这一行有 $a_{iu}=p_u$,$a_{i,p+1}=k'$(因为 $\mathbf{A}$ 是一个 $OA(p+1,n)$)。但是 u 却无法知道 n 个可能的行 i 中哪一行是正确的(即被管理者随机选中的那行)。因此只基于自己的部分信息,u 正确地猜出密钥的概率是$\dfrac{1}{n}$。

因此找到了能以较高的概率完成工作的$(2,p)$门限方案。

关于门限方案的更多内容可以参见文献[4]。

4.4.3 基于区组设计的匿名门限方案

在一个匿名(q,p)门限方案(anonymous(q,p)-threshold scheme)中,p 个人接收到信息的 p 个不同部分片段,且密钥可以从任意 q 个部分片段计算得到,而不必知道谁持有哪一部分片段。显然在 4.4.2 小节中,由正交阵列构建的门限方案不是匿名的。下面利用可解的(b,v,r,k,λ)设计来寻找匿名(q,p)门限方案,特别是匿名$(2,p)$门限方案。

假设有一个可解的(b,v,r,k,λ)设计,且 $\lambda=1$,每个区组里有 $k=p$ 个样品。设 $C_1,C_2,\cdots,C_r$ 是可解类,而每一个可解类有$\dfrac{v}{p}$个区组,每一个样品在每一个可解类中

出现一次，所以可解类的数量可由等式(4.4)计算，为

$$r=\frac{\lambda(v-1)}{p-1}=\frac{v-1}{p-1}$$

假设这个组的所有 p 个人都知道这些可解类。取可能的密钥集合 K 为$\{1,2,\cdots,r\}$，且部分信息片段的集合 P 为 V，即样品的集合。假设现在管理者想要分配密钥$k\in K$，那么他从可解类 C_k 中随机选取一个区组，并把这个区组中的 p 个片段的部分信息给这 p 个人，每个人给一个片段。注意，现在任意两个人能够确定这个密钥 k，因为如果 p_u 是给某人 u 的部分信息，p_v 是给某人 v 的部分信息，又因为 $\lambda=1$，所以这一设计中存在唯一一个包含样品 p_u 和 p_v 的区组。u 和 v 能够找到这个区组，并知道如果包含 p_u 和 p_v 的区组是在可解类 C_k 中，则这个密钥就是 k。注意，这是匿名的，因为哪个人持有哪个片段无关紧要，重要的是这两个人持有哪些片段。

下面的问题是：持有部分信息 p_u 的人(表示为 u)能确定出这一密钥的概率是多少？因为每一个可解类正好有一个包含 p_u 的区组，所以正确地猜出管理者心中的密钥的概率为$\frac{1}{r}$，其中 r 是可解类的个数。这个概率正好等于没有给出部分信息时正确猜测的概率。如果 r 非常大，那么就有一个非常安全的匿名$(2,p)$门限方案。

例如，

$$\begin{array}{llll} C_1: & \{1,2,3\} & \{4,5,6\} & \{7,8,9\} \\ C_2: & \{1,4,7\} & \{2,5,8\} & \{3,6,9\} \\ C_3: & \{1,5,9\} & \{2,6,7\} & \{3,4,8\} \\ C_4: & \{1,6,8\} & \{2,4,9\} & \{3,5,7\} \end{array}$$

是一个带有可解类 C_1,C_2,C_3,C_4 的可解的(12,9,4,3,1)设计。此设计可用于构建匿名的(2,3)门限方案。如果管理者想要分配密钥 3，那么他在 C_3 中随机选出一个区组，比如说是$\{3,4,8\}$。这个部分信息分配给这个区组中的 3 个人。例如，3 和 8 这两个人知道管理者想要的唯一区组是$\{3,4,8\}$，它在 C_3 中，所以知道密钥是 3。

4.5　注记

组合数学的内容从来都是驳杂的。本章只介绍了组合数学中一些最基本的概念和内容，有兴趣的读者可参阅文献[1]～[3]。

组合数学在信息安全中有着广泛的应用，这里只给出了几个简单的实例。有兴趣的读者可参阅文献[5]、[6]、[10]。

参考文献

[1]　(美)Brualdi R A 著. 冯舜玺等译. 组合数学. 第 4 版. 北京：机械工业出版社，2005
[2]　(美)Roberts F S 等著. 冯速译. 应用组合数学. 第 2 版. 北京：机械工业出版社，2007
[3]　(美)Ryser H J 著. 李乔译. 组合数学. 北京：科学出版社，1983
[4]　Colbourn C J，Dinitz J H(eds.). The CRC Handbook of Combinatorial Designs，CRC Press，

Boca Raton, FL, 1996

[5] Stinson D R. The Combinatorics of Authentication and Secrecy Codes, J. Cryptology, 2(1990), 23-49

[6] Stinson D R. Combinatorical Designs : Constructions and Analysis, Springer-Verlag, New York, 2003

[7] Roberts F S. Applications of Ramsey Theory, Discrete Appl. Math., 9(1984), 251-261

[8] Lindner C C. Rodger C A. Design Theory, CRC Press, Boca Raton, FL, 1997

[9] Ray-Chaudhuri D K, Wilson R M. Solution of Kirkman's schoolgirl problem, American Mathematical Society Proceedings, Symposium on Pure Mathematics, 19(1971), 187-204

[10] Dingyi Pei. Authentication codes and combinatorial designs, Taylor & Francis Group, LLC, 2006

第5章　概率论方法与技术

在信息安全尤其是在密码学中,“可能性”起着十分重要的作用。密码算法的安全性就是用可能性来衡量的,如从已知密文恢复明文的可能性。密码算法通常都是概率算法,即对同样的输入,算法输出的结果是分布在一个集合上的,涉及的集合通常都是离散的。因此,本章主要介绍研究可能性的理论——概率论,并主要关注概率空间为离散集合的情形。本章首先给出事件、样本空间和概率的定义,然后给出随机变量的概念,讨论一些基本概念和性质,如期望、方差的定义和计算等。随后,讨论几个典型分布,如二项分布、泊松分布、正态分布等,并进一步给出了刻画它们之间关系的大数定律、中心极限定理等,以及几个常用和重要的不等式。最后,用一个例子说明概率论方法与技术在密码学中的基本应用技巧。

5.1　事件、样本空间和概率

当对事务进行研究,试图形成一些理论时,最基础的工作是要对研究对象进行分类、概括,形成概念,即进行抽象化或理想化。也就是说,要建立理想模型。概率理论的第一个理想化是关于实验或观察的可能结果。

事件　把实验或观察的结果叫做事件。事件分为复合事件和简单事件。复合事件是一些简单事件的集合。在一般的概率论中,简单事件是最基本的概念,就像几何中的点一样是不定义的。在具体应用中,简单事件是由具体情景来规定的,代表可以想象的结果,用它们来定义理想的实验。在密码学中,经常把算法的一个可能输出定义为简单事件。

样本空间　简单事件也称为样本点。所有样本点的全体称为样本空间。最多含有可数个样本点的样本空间称为离散样本空间。

事件之间的关系及运算　给定一个样本空间 Ω,用大写字母表示事件,即样本点的集合。点 ω 包含在事件 A 中,用 $\omega\in A$ 表示。从给定的一些事件出发,经过某些变换可以形成一些新事件。这些变换,可以用逻辑的语言表述为“或”、“与”、“非”,而用集合论的语言可以表述为“并”、“交”、“补”。

并　对于集合 A 与 B,称 $A\cup B=\{\omega\in\Omega\mid\omega\in A \text{ 或 } \omega\in B\}$ 为集合 A 与 B 的并,表示由属于集合 A 或 B 的点形成的集合。用概率论的语言,$A\cup B$ 表示事件“事件 A 与 B 至少发生一个”。

交　称 AB 或 $A\cap B=\{\omega\in\Omega\mid\omega\in A \text{ 且 } \omega\in B\}$ 为两个集合 A 与 B 的交,表示由既属于集合 A 同时又属于集合 B 的点形成的集合。用概率论的语言,AB 表示事件“事件 A 与 B 同时发生”。当 A 与 B 不相交时,称 A 与 B 互不相容。

补　如果 A 是 Ω 的子集,则称 $\overline{A}=\{\omega\in\Omega\mid\omega\notin A\}$ 为 A 的补,表示 Ω 中不属于 A 的点的集合。用概率论的语言,$\overline{A}$ 表示事件“事件 A 不发生”。

差　属于 B 但不属于 A 的点的集合称为 B 与 A 的差，记做 $B\backslash A$。用概率论的语言，表示事件"事件 B 发生，同时 A 不发生"。

不可能事件和必然事件　空集 $\varnothing$ 表示不可能事件，而集合 Ω 称为必然事件。

上述定义可以推广到任意有限个事件的情形。

概率　给定一个离散样本空间 $\Omega=\{\omega_i \mid i\in N\}$，假定对每个点 ω_i 都赋予一个非负实数，这个数称为 ω_i 的概率，记为 $\Pr[\omega_i]$。这些数必须满足

$$\sum_{i\in N}\Pr[\omega_i]=1 \tag{5.1}$$

规定 $\Pr[\varnothing]=0$。任何事件 A 的概率是 A 中所包含样本点概率的总和，即

$$\Pr[A]=\sum_{\omega_i\in A}\Pr[\omega_i] \tag{5.2}$$

经常遇到的一个特殊情形是每个样本点的概率相等。

容易得到下面的定理。

定理 5.1.1　对给定样本空间 Ω 和事件 $A,B\subseteq\Omega$，有

$$\Pr[A\cup B]=\Pr[A]+\Pr[B]-\Pr[AB] \tag{5.3}$$

特别地，如果 A 与 B 是互不相容的，则

$$\Pr[A\cup B]=\Pr[A]+\Pr[B]$$

上面的定理可以推广到任意有限个事件的情形。

定理 5.1.2　对给定样本空间 Ω 和事件 $A_1,\cdots,A_n\subseteq\Omega$，有

$$\Pr[A_1\cup\cdots\cup A_n]=\sum_{i=1}^{n}\Pr[A_i]-\sum_{\substack{i,j=1\\ i<j}}^{n}\Pr[A_iA_j]+\sum_{\substack{i,j,k=1\\ i<j<k}}^{n}\Pr[A_iA_jA_k]-\cdots+(-1)^{n-1}\Pr[A_1A_2\cdots A_n] \tag{5.4}$$

特别地，如果 $A_1,\cdots,A_n$ 是两两互不相容的，则

$$\Pr[A_1\cup\cdots\cup A_n]=\sum_{i=1}^{n}\Pr[A_i] \tag{5.5}$$

证明：运用容斥原理即得。

古典概型　对于有限样本空间 Ω，定义每个样本点的概率都一样。这时，事件 A 的概率就是

$$\Pr[A]=\frac{|A|}{|\Omega|} \tag{5.6}$$

这里，记号 $|B|$ 表示集合 B 所含点的个数。

例 5.1.1　有序样本　考虑集合 $S=\{a_1,\cdots,a_n\}$。任何 r 个元素的有序排列 $(a_{j_1},\cdots,a_{j_r})$ 称为大小为 r 的一个有序样本。这样，就有一个样本空间 $\Omega=\{(a_{j_1},\cdots,a_{j_r})\mid a_{j_i}\in S\}$，易知 $|\Omega|=n^r$。按照古典概型认为，这些样本点的概率是一样的。要问样本中的元素互不相同这样的事件的概率有多大？即事件 $A=\{(a_{j_1},\cdots,a_{j_r})\mid a_{j_i}\neq a_{j_k},j\neq k\}$ 的概率如何？显然，$|A|=n(n-1)\cdots(n-r+1)=(n)_r$。因此

$$\Pr[A]=\frac{(n)_r}{n^r}=\left(1-\frac{1}{n}\right)\left(1-\frac{2}{n}\right)\cdots\left(1-\frac{r-1}{n}\right) \tag{5.7}$$

这个例子有一个常见的解释。假设某一班中有 r 个学生，每个学生的生日在一

年的 365 天中是等可能的。问"r 个学生中至少有两人的生日在同一天"的概率有多大？根据式(5.7)，这个概率为

$$\Pr[\overline{A}] = 1 - \Pr[A] = 1 - \frac{(365)_r}{365^r}$$

在 $r=23$ 时，上式的值为 0.5073。即：当一个班有 23 名学生时，至少有两个人的生日在同一天的概率就超过了 1/2。上述原理在密码学中被用来估计找到杂凑函数的碰撞消息的概率。由于上面的解释，这种方法被称为生日攻击。

5.2　条件概率和独立性

条件概率　条件概率是概率论中的一个基本工具。对于一个给定条件 A，考虑各种事件的条件概率相当于把 A 看作一个新的样本空间。为了使新的样本空间的总概率为 1，需将原空间中所有事件的概率都乘上因子 $\frac{1}{\Pr[A]}$。这说明，所有概率的一般定理，对在任何给定条件下的条件概率依然成立。

定义 5.2.1　设 $\Pr[A]>0$。称

$$\frac{\Pr[AB]}{\Pr[A]} \tag{5.8}$$

为在事件 A 的条件下，事件 B 的条件概率，记为 $\Pr[B|A]$。

全概率公式和乘法公式　考虑样本空间 Ω 的一个划分，即一组两两互不相容事件 $A_1,\cdots,A_n$，满足 $A_1\cup\cdots\cup A_n=\Omega$。显然，

$$B = BA_1 \cup \cdots \cup BA_n$$

因此，由定理 5.1.2，有

$$\Pr[B] = \sum_{i=1}^{n} \Pr[BA_i] \tag{5.9}$$

由条件概率的定义 5.2.1，有

$$\Pr[BA_i] = \Pr[B \mid A_i]\Pr[A_i] \tag{5.10}$$

于是，有全概率公式

$$\Pr[B] = \sum_{i=1}^{n} \Pr[B \mid A_i]\Pr[A_i] \tag{5.11}$$

由条件概率的定义 5.2.1 知，当 $\Pr[A]>0$ 时，有

$$\Pr[AB] = \Pr[B \mid A]\Pr[A] \tag{5.12}$$

该式称为**乘法公式**。用数学归纳法可将其推广：假设事件 $A_1,\cdots,A_n$ 满足条件 $\Pr[A_1\cdots A_n]>0$，则

$$\Pr[A_1\cdots A_n] = \Pr[A_1]\Pr[A_2 \mid A_1]\cdots\Pr[A_n \mid A_1\cdots A_{n-1}] \tag{5.13}$$

贝叶斯公式　设 $\Pr[A]>0$，$\Pr[B]>0$。由乘法公式(5.12)，有

$$\Pr[AB] = \Pr[B \mid A]\Pr[A] = \Pr[A \mid B]\Pr[B] \tag{5.14}$$

由此得到贝叶斯公式

$$\Pr[A \mid B] = \frac{\Pr[A]\Pr[B \mid A]}{\Pr[B]} \tag{5.15}$$

贝叶斯定理 如果事件组 $A_1,\cdots,A_n$ 是 Ω 的一个划分，则由全概率公式(5.11)，有

$$\Pr[A_i \mid B] = \frac{\Pr[A_i]\Pr[B \mid A_i]}{\sum_{j=1}^{n} \Pr[A_j]\Pr[B \mid A_j]} \tag{5.16}$$

例 5.2.1 假设匣中有两枚硬币：A_1 为对称的硬币，“正面”Z 出现的概率等于 1/2；而 A_2 为不对称的硬币，“正面”Z 出现的概率等于 1/3。随意选出一枚硬币，并将其投掷，结果出现正面。问选到硬币为对称硬币的概率多大？

首先建立概率模型。选取样本空间 $\Omega=\{A_1Z,A_1F,A_2Z,A_2F\}$ 用以描绘选取和投掷的结果，其中 A_1Z 表示“选中硬币 A_1 并掷出正面 Z”等，F 表示硬币掷出反面。根据假设，有

$$\Pr[A_1] = \Pr[A_2] = \frac{1}{2}$$

和

$$\Pr[Z \mid A_1] = \frac{1}{2}, \quad \Pr[Z \mid A_2] = \frac{1}{3}$$

由式(5.12)，这些条件就唯一确定了各结果的概率：

$$\Pr[A_1Z] = \frac{1}{4}, \quad \Pr[A_1F] = \frac{1}{4}, \quad \Pr[A_2Z] = \frac{1}{6}, \quad \Pr[A_2F] = \frac{1}{3}$$

由贝叶斯定理，“选到硬币为对称硬币”的概率为

$$\Pr[A_1 \mid Z] = \frac{\Pr[A_1]\Pr[Z \mid A_1]}{\Pr[A_1]\Pr[Z \mid A_1] + \Pr[A_2]\Pr[Z \mid A_2]} = \frac{3}{5}$$

独立性 独立性概念在一定意义上对概率论来说具有核心作用。独立性概念决定了概率论的特色，使概率论不同于研究集合的大小。

一般情况下，条件概率 $\Pr[B|A]$ 不等于“绝对”概率 $\Pr[B]$。

定义 5.2.2 称事件 A 和 B 是独立的或统计独立的，如果

$$\Pr[AB] = \Pr[A]\Pr[B] \tag{5.17}$$

当 A 和 B 是独立的，则有

$$\Pr[B \mid A] = \Pr[B]$$

对于多个事件，可以有更强的独立性。

定义 5.2.3 称集合 $A_1,\cdots,A_n$ 全体独立或全体统计独立，如果对于任何 $k=1,2,\cdots,n$ 和 $1\leqslant i_1<\cdots<i_k\leqslant n$，有

$$\Pr[A_{i_1}\cdots A_{i_k}] = \Pr[A_{i_1}]\cdots\Pr[A_{i_k}]$$

一般说来，两两独立的事件未必全体独立。

乘积空间 设 Ω 和 Ξ 是两个样本空间，考虑它们的直积 $\Omega\times\Xi=\{(\omega,\theta)\mid\omega\in\Omega,\theta\in\Xi\}$。称 $\Omega\times\Xi$ 为 Ω 和 Ξ 的乘积空间，如果每个样本点 (ω,θ) 的概率定义为

$$\Pr[(\omega,\theta)] = \Pr[\omega]\Pr[\theta]$$

乘积空间 $\Omega\times\Xi$ 的概率论解释为：具有样本空间 Ω 和 Ξ 的两个相继的独立实验的样本空间。

上述概念可以推广到 n 个相继的独立试验的情形。由此，我们说重复的独立试验，意指由相同的样本空间做成的乘积样本空间中的样本点。

对于 $\Omega\times\Xi$ 中形如 (A,B) 的事件，其中 A、B 分别是 Ω、Ξ 中的事件，显然有

$$\Pr[(A,B)]=\Pr[A]\Pr[B]$$

例 5.2.2　给出一个典型的重复独立实验：**伯努利试验序列**。

在重复的独立试验中，如果每次试验仅有两个可能结果，而且其相应的概率在每次试验中都是相同的，则称这一串重复的独立试验是伯努利试验序列。

用 0、1 记载每次试验的结果，用 p 表示 1 出现的概率，q 表示 0 出现的概率。n 次伯努利试验的样本空间 Ω 含有长为 n 的 0，1 串组成的 2^n 个样本点，即 $\Omega=\{\omega\mid\omega=(a_1,\cdots,a_n),a_i\in\{0,1\}\}$。因为试验是独立的，所以每个串出现的概率等于各个分量的概率的乘积，即

$$\Pr[\omega]=p^{\sum a_i}q^{1-\sum a_i}$$

5.3　随机变量、期望值和方差

上节讨论了各种事件的概率，其实事件和事件空间的自然本性并不重要，重要的是某种数字特征，其值依赖于基本事件。随机变量的概念就抽象出了人们关心的结果的数值特征。

随机变量　任一定义在样本空间上的实值函数称为随机变量。令 X 为随机变量，x 为 X 的一个取值。所有使得 X 取值 x 的样本点构成事件，记为 $X=x$，其概率记为 $\Pr[X=x]$，称为随机变量 X 取值 x 的概率。

分布　函数 $p(x):=\Pr[X=x]$ 叫做 X 的（概率）分布。严格地说，集合 $\{p(x)\}$ 叫做 X 的（概率）分布，而 $p(x)$ 叫做该分布的密度函数。实际上，无论是分布还是密度函数都是指的同一件事情，只不过有整体和局部的不同。因此，在不引起不便的前提下，在本书中不严格区分这两个概念，而含混地称为分布。

随机向量与联合分布　考虑定义在同一个样本空间中的两个随机变量 X 和 Y，向量 $(\boldsymbol{X},\boldsymbol{Y})$ 称为随机向量。同时满足 $X=x$ 和 $Y=y$ 这两个条件的样本点的全体构成一个事件，其概率记做 $\Pr[X=x,Y=y]$。函数 $p(x,y):=\Pr[X=x,Y=y]$ 叫做 X、Y 的联合（概率）分布。对一个分量求和，就得到关于另一个变量的分布，叫做边缘分布，如 $p(x):=\sum\limits_y\Pr[X=x,Y=y]$ 就是 X 的分布。随机向量和联合分布这两个概念完全可以推广到多个随机变量的系统上去。

独立随机变量　设 $X_1,\cdots,X_n$ 为随机变量，如果对任意 $x_1,\cdots,x_n$，有

$$\Pr[X_1=x_1,\cdots,X_n=x_n]=\Pr[X_1=x_1]\cdots\Pr[X_n=x_n] \tag{5.18}$$

则称随机变量 $X_1,\cdots,X_n$ 为（全体）独立的。

期望　对随机变量取值的概率加权平均就得到期望的概念。

定义 5.3.1 设 X 为一个随机变量，其可能取的值为 $x_1, x_2, \cdots, x_k$，若级数

$$E(X) := \sum x_k \Pr[X = x_k] \tag{5.19}$$

绝对收敛，则定义这个级数的值为 X 的期望。这时就说 X 有有限的期望。如果 $\sum |x_k| \Pr[X = x_k]$ 发散，则说 X 没有有限的期望。

定理 5.3.1 对任一函数 $\varphi(x)$，定义一个新的随机变量 $\varphi(X)$。如果 $\varphi(X)$ 具有有限期望，则

$$E(\varphi(X)) = \sum \varphi(x_k) \Pr[X = x_k] \tag{5.20}$$

此处的级数绝对收敛当且仅当 $E(\varphi(X))$ 存在。对任一常数 a，有 $E(aX) = aE(X)$。

证明：由期望的定义，有 $E(\varphi(X)) := \sum \varphi(x_k) \Pr[\varphi(X) = \varphi(x_k)]$。而 $\Pr[\varphi(X) = \varphi(x_k)] = \sum\limits_{x_k \in \varphi^{-1}\varphi(x_k)} \Pr[X = x_k]$。从而定理成立。

定理 5.3.2 如果 $X_1, \cdots, X_n$ 都是具有有限期望的随机变量，则它们的和的期望存在，且其和的期望就等于期望的和：

$$E(X_1 + \cdots + X_n) = E(X_1) + \cdots + E(X_n) \tag{5.21}$$

证明：只需对 $n=2$ 进行证明。由期望的定义，有

$$\begin{aligned}
E(X_1 + X_2) &= \sum_z z \Pr[X_1 + X_2 = z] \\
&= \sum_z z \sum_{\substack{x_1, x_2 \\ x_1 + x_2 = z}} \Pr[X_1 = x_1, X_2 = x_2] \\
&= \sum_z (x_1 + x_2) \sum_{\substack{x_1, x_2 \\ x_1 + x_2 = z}} \Pr[X_1 = x_1, X_2 = x_2] \\
&= \sum_z x_1 \sum_{\substack{x_1, x_2 \\ x_1 + x_2 = z}} \Pr[X_1 = x_1, X_2 = x_2] \\
&\quad + \sum_z x_2 \sum_{\substack{x_1, x_2 \\ x_1 + x_2 = z}} \Pr[X_1 = x_1, X_2 = x_2] \\
&= \sum_{x_1} x_1 \sum_{\substack{z, x_2 \\ z - x_2 = x_1}} \Pr[X_1 = x_1, X_2 = x_2] \\
&\quad + \sum_{x_2} x_2 \sum_{\substack{x_1, z \\ z - x_1 = x_2}} \Pr[X_1 = x_1, X_2 = x_2] \\
&= \sum_{x_1} x_1 \sum_{x_2} \Pr[X_1 = x_1, X_2 = x_2] \\
&\quad + \sum_{x_2} x_2 \sum_{x_1} \Pr[X_1 = x_1, X_2 = x_2] \\
&= E(X_1) + E(X_2)
\end{aligned}$$

定理 5.3.3 如果 X、Y 是具有有限期望的相互独立的随机变量，则它们的积也

是具有有限期望的随机变量,且

$$E(XY) = E(X)E(Y) \tag{5.22}$$

证明:留给读者。

方差　随机变量的方差和标准差表征了随机变量取值的散布程度。$E(X^t)$称为 X 的 t 阶矩。如果 X 的二阶矩是存在的,则可以有下面的定义。

定义 5.3.2　称

$$\mathrm{Var}(X) := E((X - E(X))^2) = E(X^2) - (E(X))^2 \tag{5.23}$$

为随机变量 X 的方差。称 $\sigma := \sqrt{\mathrm{Var}(X)}$ 为标准差。

定理 5.3.4　如果 X、Y 是具有有限期望和方差的相互独立的随机变量,则

$$\mathrm{Var}(X + Y) = \mathrm{Var}(X) + \mathrm{Var}(Y) \tag{5.24}$$

证明:设 $E(X)=\mu_X$,$E(Y)=\mu_Y$。因为 X、Y 独立,由定理 5.3.3,有 $E(XY)=\mu_X\mu_Y$。于是

$$\begin{aligned}\mathrm{Var}(X + Y) &= E((X + Y)^2) - (\mu_X + \mu_Y)^2 \\ &= E(X^2 + 2XY + Y^2) - (\mu_X^2 + 2\mu_X\mu_Y + \mu_Y^2) \\ &= E(X^2) + 2E(XY) + E(Y^2) - \mu_X^2 - 2\mu_X\mu_Y - \mu_Y^2 \\ &= E(X^2) - \mu_X^2 + E(Y^2) - \mu_Y^2 + 2E(XY) - 2\mu_X\mu_Y \\ &= \mathrm{Var}(X) + \mathrm{Var}(Y)\end{aligned}$$

另一方面,当 X、Y 相关时,$\mathrm{Var}(X+Y)$与 $\mathrm{Var}(X)+\mathrm{Var}(Y)$可能不同。

定理 5.3.5　令 a、b 为常量。则

$$\mathrm{Var}(aX + b) = a^2\mathrm{Var}(X) \tag{5.25}$$

证明:设 $E(X)=\mu$。注意到 $E(aX+b)=aE(X)+b$。于是

$$\begin{aligned}\mathrm{Var}(aX + b) &= E((aX + b)^2) - (E(aX + b))^2 \\ &= E(a^2X^2 + 2abX + b^2) - (a\mu + b)^2 \\ &= a^2E(X^2) + 2abE(X) + b^2 - (a^2\mu^2 + 2ab\mu + b^2) \\ &= a^2(E(X^2) - \mu^2) \\ &= a^2\mathrm{Var}(X)\end{aligned}$$

例 5.3.1　设 X 是伯努利随机变量,以概率 p 和 q 取值 1 和 0,则 X 的期望

$$E(X) = 1 \times \Pr[X = 1] + 0 \times \Pr[X = 0] = p \tag{5.26}$$

X 的方差

$$\begin{aligned}\mathrm{Var}(X) &= E((X - E(X))^2) = E((X - p)^2) \\ &= (1 - p)^2p + p^2q = pq \end{aligned} \tag{5.27}$$

例 5.3.2　设 $X_1,\cdots,X_n$ 是独立的与例 5.3.1 同分布的伯努利随机变量。设 $S_n=X_1+\cdots+X_n$。则 S_n 的期望

$$E(S_n) = E(X_1) + \cdots + E(X_n) = np \tag{5.28}$$

S_n 的方差

$$\mathrm{Var}(S_n) = \mathrm{Var}(X_1) + \cdots + \mathrm{Var}(X_n) = npq \tag{5.29}$$

5.4 二项分布、泊松分布和正态分布

本节将围绕伯努利试验给出一些典型的分布，这些分布在概率论中有着重要的历史和现实意义。用这些分布之间的逼近关系，还例示了在概率论中具有中心意义的大数定律和中心极限定理。

在例 5.2.2 中，给出了伯努利试验序列的定义。现在可以用随机变量及其分布的概念描述伯努利试验序列。

伯努利试验仅有两个可能结果，一串重复的独立伯努利试验就是伯努利试验序列。如果用 0、1 记载每次试验的结果，用 p 表示 1 出现的概率，q 表示 0 出现的概率，就定义了一个取值 0、1 的随机变量。用 X_k 记第 k 次试验的随机变量。n 次伯努利试验就对应着 n 个随机变量 $X_1,\cdots,X_n$ 的联合分布。

一致分布 对于由长为 n 的 0、1 串组成的样本，用 U_n 表示取每个串的概率为 2^{-n} 的随机变量，即其取每个串的概率都是一样的，其分布函数称为一致分布。这里随机变量的概念稍有扩张，其取值不是在实数域中，而是在 0、1 串上。实际上，一致分布就是伯努利试验序列 $p=1/2$ 的情形，此时，U_n 就是随机向量 $(X_1,\cdots,X_n)$。

二项分布 在 n 次伯努利试验中，人们往往只关心 1 出现的个数。令 $S_n=X_1+\cdots+X_n$，则 S_n 就是取值为 1 出现个数的随机变量。记 $b(k;n,p)=\Pr[S_n=k]$，则 $b(k;n,p)$ 就是 S_n 的分布函数，称为二项分布。易知，$b(k;n,p)=\binom{n}{k}p^kq^{n-k}$。

古典大数定律 对于概率，有这样一种直观的概念：如果在 n 次相同的试验中，A 发生 k 次，那么当 n 很大时，$\frac{k}{n}$ 将接近 A 的概率 p。

把这件事精确化，尝试用伯努利试验来描述推导相应的结论。n 次相同的试验理解为成功概率为 p 的伯努利试验序列。$\frac{S_n}{n}$ 就是成功的平均次数而且接近 p。考虑 $\frac{S_n}{n}$ 超过 $p+\varepsilon(\varepsilon>0)$ 的概率，即 $\Pr\left[\frac{S_n}{n}>p+\varepsilon\right]$。

当 $r>np$ 时，有

$$\Pr[S_n\geqslant r]=\sum_{v\geqslant 0}b(r+v;n,p)\leqslant b(r;n,p)\frac{rq}{r-np}$$

这是因为此时有

$$\begin{aligned}\frac{b(r+v;n,p)}{b(r+v-1;n,p)}&=\frac{(n-r-v+1)p}{(r+v)q}\\&=1-\frac{r+v-(n+1)p}{(r+v)q}\leqslant 1-\frac{r-(n+1)p}{rq}\\&\leqslant 1-\frac{r-np}{rq}\end{aligned}$$

另一方面，

$$1 \geqslant \sum_{k=\lfloor (n+1)p \rfloor}^{r} b(k;n,p) \geqslant (r-np)b(r;n,p)$$

从而当 $r>np$ 时，有

$$\Pr[S_n \geqslant r] \leqslant \frac{rq}{(r-np)^2}$$

于是，

$$\Pr\left[\frac{S_n}{n} > p+\varepsilon\right] = \Pr[S_n > n(p+\varepsilon)] \leqslant \frac{(p+\varepsilon)q}{n\varepsilon^2}$$

因此，当 n 增大时，

$$\Pr\left[\frac{S_n}{n} > p+\varepsilon\right] \to 0$$

类似可得，$\Pr\left[\frac{S_n}{n} < p-\varepsilon\right] \to 0$。最后，有

$$\Pr\left[\left|\frac{S_n}{n} - p\right| < \varepsilon\right] \to 1$$

这就是大数定律。大数定律反映了后天试验的频率应该接近先验概率。

上面是用分析的方法给出了古典大数定律的证明。下一节可以用概率不等式给出大数定律的一个更简洁的证明。

泊松分布　概率分布函数 $p(k;\lambda)=\mathrm{e}^{-\lambda}\frac{\lambda^k}{k!}$称为泊松分布。

二项分布的泊松逼近　对于 n 大，p 小，而乘积 $\lambda=np$ 大小适中的二项分布，可以用泊松分布逼近。首先，对于 $k=0$，有

$$b(0;n,p) = (1-p)^n = \left(1-\frac{\lambda}{n}\right)^n$$

取对数并利用泰勒级数展开，有

$$\lg b(0;n,p) = n\lg\left(1-\frac{\lambda}{n}\right) = -\lambda - \frac{\lambda^2}{2n} - \cdots$$

因此，对于充分大的 n，有

$$b(0;n,p) \approx \mathrm{e}^{-\lambda}$$

这里，≈表示相差阶为 n^{-1} 的无穷小量。其次，对任一固定 k 和充分大 n，有

$$\frac{b(k;n,p)}{b(k-1;n,p)} = \frac{\lambda-(k-1)p}{kq} = \frac{\lambda-\lambda p}{kq} + \frac{\lambda p-(k-1)p}{kq} \approx \frac{\lambda}{k}$$

于是，最终有

$$b(k;n,p) \approx \frac{\lambda}{k} b(k-1;n,p) \approx \frac{\lambda}{k}\frac{\lambda}{k-1} b(k-2;n,p)$$

$$\approx \cdots \approx \frac{\lambda^k}{k!}\mathrm{e}^{-\lambda} = p(k;\lambda)$$

正态分布　函数

$$\varphi(x) = \frac{1}{\sqrt{2\pi}}\mathrm{e}^{-\frac{1}{2}x^2}$$

称为正态密度函数，它的积分

$$\Phi(x) = \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{x} e^{-\frac{1}{2}y^2}\,dy$$

就叫做正态分布函数。

正态分布的密度函数是关于 y 轴对称的平缓的钟形曲线，正态分布也可用来逼近二项分布。

迪莫弗-拉普拉斯极限定理 对固定的 z_1 和 z_2，当 $n\to\infty$时，有

$$\Pr\left[z_1 \leqslant \frac{S_n - np}{\sqrt{npq}} \leqslant z_2\right] \to \Phi(z_2) - \Phi(z_1)$$

注意： ① 迪莫弗-拉普拉斯极限定理讲的是正态分布对二项分布的主要部分的逼近。而二项分布的泊松逼近是对二项分布的每一项用泊松分布在特殊情形下的逼近。

② 古典大数定律和迪莫弗-拉普拉斯极限定理是 5.5 节的一般大数定律和中心极限定理情形为伯努利试验时的特例。

5.5 大数定律和中心极限定理

首先给出一般的大数定律和中心极限定理。然后给出一些特殊的更精确的大数定律。

大数定律 设$\{X_k\}$是相互独立且具有公共分布的随机变量序列。如果其期望$\mu=E(X_k)$存在，并令 $S_n=X_1+\cdots+X_n$，则对每个 $\varepsilon>0$，当 $n\to\infty$总有

$$\Pr\left[\left|\frac{S_n}{n} - \mu\right| > \varepsilon\right] \to 0 \tag{5.30}$$

即平均数 S_n/n 与期望 μ 之间的偏差小于任意给定的 ε 的概率趋于 1。

中心极限定理 设$\{X_k\}$是相互独立且具有公共分布的随机变量序列。假定 $\mu=E(X_k)$和 $\sigma^2=\mathrm{Var}(X_k)$都存在，并令 $S_n=X_1+\cdots+X_n$，则对每个固定的 β，当 $n\to\infty$总有

$$\Pr\left[\frac{S_n - n\mu}{\sigma\sqrt{n}} < \beta\right] \to \Phi(\beta) \tag{5.31}$$

其中 $\Phi(\beta)$是正态分布函数。

在这里不打算给出大数定律和中心极限定理的证明。读者可以参考任何一本概率论的标准教程。

注意： 式(5.31)比式(5.30)强，这是因为式(5.31)给出了偏差 $|n^{-1}S_n-\mu|>\sigma/\sqrt{n}$的概率的估计值。另一方面，大数定律并不要求 X_k 具有有限方差，从这种意义上讲，它比中心极限定理更一般。

上述大数定律和极限定理都没有给出精确的估计。借助一些概率工具，如切比雪夫不等式等，可以得到一些更精确的大数定律(在较强的假设下，也以不等式的形式出现)。这些不等式本身也是非常有用的，值得单独介绍。

定理 5.5.1(马尔科夫不等式) 设 X 是非负随机变量。则对任意 $\varepsilon>0$，有

$$\Pr[X \geqslant \varepsilon] \leqslant \frac{E(X)}{\varepsilon} \tag{5.32}$$

证明：$E(X)=\sum_{x} x\Pr[X=x]=\sum_{x<\varepsilon} x\Pr[X=x]+\sum_{x\geqslant\varepsilon} x\Pr[X=x]$

$$\geqslant \sum_{x\geqslant\varepsilon} x\Pr[X=x]\geqslant \varepsilon\sum_{x\geqslant\varepsilon}\Pr[X=x]=\varepsilon\Pr[X\geqslant\varepsilon]$$

定理 5.5.2（切比雪夫不等式） 设 X 是任一随机变量，其期望和方差都存在。则对任意 $\varepsilon>0$，有

$$\Pr[|X-E(X)|\geqslant\varepsilon]\leqslant\frac{\mathrm{Var}(X)}{\varepsilon^2} \tag{5.33}$$

证明：令 $Y=(X-E(X))^2$。利用马尔科夫不等式

$$\begin{aligned}\Pr[|X-E(X)|\geqslant\varepsilon]&=\Pr[(X-E(X))^2\geqslant\varepsilon^2]\\&=\Pr[Y\geqslant\varepsilon^2]\leqslant\frac{E(Y)}{\varepsilon^2}=\frac{\mathrm{Var}(X)}{\varepsilon^2}\end{aligned}$$

定理 5.5.3 设$\{X_k\}$是相互独立且具有公共期望 μ 和方差 σ^2 的随机变量序列。则对任 $\varepsilon>0$，有

$$\Pr\left[\left|\frac{1}{n}\sum_{k=1}^{n}X_k-\mu\right|\geqslant\varepsilon\right]\leqslant\frac{\sigma^2}{n\varepsilon^2} \tag{5.34}$$

证明：令 $S_n=\frac{1}{n}\sum_{k=1}^{n}X_k$。由定理 5.3.4 和定理 5.3.5，有

$$\mathrm{Var}(S_n)=\frac{\mathrm{Var}(X_1)+\cdots+\mathrm{Var}(X_n)}{n^2}=\frac{\sigma^2}{n}$$

由切比雪夫不等式，有

$$\Pr[|S_n-\mu|\geqslant\varepsilon]\leqslant\frac{\mathrm{Var}(S_n)}{\varepsilon^2}=\frac{\sigma^2}{n\varepsilon^2}$$

5.4 节中的古典大数定律显然是定理 5.5.3 的推论。

上述结果可推广到期望和方差不同的两两独立的随机变量$\{X_k\}_{k=1}^{n}$上。此时，令 $\mu=\frac{1}{n}\sum_{k=1}^{n}E(X_k)$，有

$$\Pr\left[\left|\frac{1}{n}\sum_{k=1}^{n}X_k-\mu\right|\geqslant\varepsilon\right]\leqslant\frac{\max_k\mathrm{Var}(X_k)}{n\varepsilon^2} \tag{5.35}$$

定理 5.5.4（Chernoff/Hoefding 界） 设$\{X_k\}$是具有期望的、全独立的随机变量，且 $X_k\in[a,b]$。则对任 $\varepsilon>0$，有

$$\Pr\left[\left|\frac{1}{n}\sum_{k=1}^{n}X_k-\frac{1}{n}\sum_{k=1}^{n}E(X_k)\right|\geqslant\varepsilon\right]\leqslant 2\mathrm{e}^{-\frac{2\varepsilon^2}{(b-a)^2}n} \tag{5.36}$$

特别，当 $X_k\in[0,1]$时，有

$$\Pr\left[\left|\frac{1}{n}\sum_{k=1}^{n}X_k-\frac{1}{n}\sum_{k=1}^{n}E(X_k)\right|\geqslant\varepsilon\right]\leqslant 2\mathrm{e}^{-2\varepsilon^2 n} \tag{5.37}$$

这个不等式的证明需要一些复杂的技巧，在这里不给出证明。读者可参阅文献[4]第2 章第 2 节。

5.6 应用举例

收缩序列是一类已证明具有较长的周期、较高的线性复杂度和良好的统计特性的序列，似乎是一类良好的密钥流生成器。但不少文献中都指出了这类生成器存在的安全缺陷，这些分析表明，收缩序列体制可以部分地被破译。本节分析收缩序列的一个方法，以说明概率论方法与技术在密码分析中的应用。该方法的基本思路是：首先进行初步理论统计分析，其次构造出与输入序列有较高符合率的拟合序列，最后使用快速相关攻击方法可以部分地破译这种体制。因为这里只是为了说明概率论方法与技术的应用，所以只介绍如何完成前两步。

5.6.1 收缩生成器的描述

收缩生成器由两个 LFSR 构成。通过用一个 $LFSR_1$ 选择另一个 $LFSR_2$ 的输出来生成密钥流，参见图 5.1。

输入：参数：两个 $LFSR\langle f_i(x), L_i\rangle, i=1,2$，

密钥：两个 LFSR 的初始状态 $a_0^{(1)}, a_0^{(2)}$。

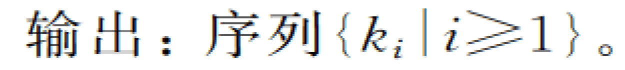
对 $i=1,2,\cdots$，完成下列步骤（记 $LFSR_i$ 的生成序列是 $y_0^{(i)}, y_1^{(i)}, \cdots, y_j^{(i)}, \cdots$）

（1）移位 $LFSR_1$ 并产生 $y_i^{(1)}$；

（2）移位 $LFSR_2$ 并产生 $y_i^{(2)}$；

（3）如果 $y_i^{(1)}=1$，则置 $k_i=y_i^{(2)}$；

如果 $y_i^{(1)}=0$，则删去 $y_i^{(2)}$。



输出：序列 $\{k_i \mid i\geqslant 1\}$。

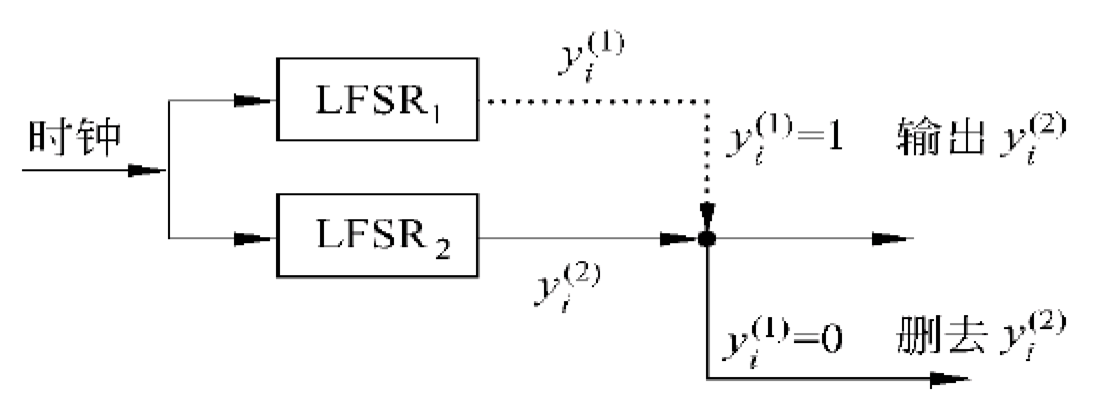


图 5.1 收缩生成器

5.6.2 收缩序列的初步理论统计分析

将 $LFSR_1$ 产生的序列即输入序列记为 $\underline{a}=(a_0, a_1, \cdots,)$，$LFSR_2$ 产生的序列即控制序列记为 $\underline{s}=(s_0, s_1, \cdots,)$，最后产生的收缩序列记为 $\underline{E}=(E_0, E_1, \cdots,)$。这里假定 $\underline{a}$ 和 $\underline{s}$ 都是 m 序列，$LFSR_2$ 的级数为 n。

对控制序列 $\underline{s}$，设在其一个首尾相接的周期段中，具有前 $i(i=0,1,\cdots,n-1)$ 比特全为 0，第 $i+1$ 比特为 1 这样特性的 n 长截段出现的概率分别为 $p_0, p_1, \cdots, p_{n-1}$，则有

$$p_i = \frac{2^{n-i-1}}{2^n - 1} \approx \frac{1}{2^{i+1}}, \quad i = 0,1,\cdots,n-1 \tag{5.38}$$

用“$E_i \downarrow a_j$”表示 E_i 取自 a_j，则有

$$p(E_0 \downarrow a_0) = p_0$$
$$p(E_0 \downarrow a_1) = p_1$$
$$\vdots$$
$$p(E_0 \downarrow a_{n-1}) = p_{n-1}$$

$$
\begin{aligned}
&p(E_1 \downarrow a_1) = p_0^2 \\
&p(E_1 \downarrow a_2) = p_0 p_1 + p_1 p_0 \\
&\quad\vdots \\
&p(E_1 \downarrow a_i) = p_0 p_{i-1} + p_1 p_{i-2} + \cdots + p_{i-1} p_0 \quad 1 \leqslant i \leqslant 2n-1 \\
&\quad\vdots \\
&p(E_1 \downarrow a_{2n-1}) = p_{n-1}^2 \\
&p(E_2 \downarrow a_2) = p_0^3 \\
&p(E_2 \downarrow a_3) = p_0 p_0 p_1 + p_0 p_1 p_0 + p_1 p_0 p_0 \\
&\quad\vdots \\
&p(E_k \downarrow a_s) = \sum_{i_0+i_1+\cdots+i_{k-1}=s-k} p_{i_0} p_{i_1} \cdots p_{i_{k-1}}
\end{aligned} \tag{5.39}
$$

由式(5.39)及 $\sum_{i_0+i_1+\cdots+i_{k-1}=s-k} 1 = C_{s-1}^{s-k}$ 可知，

$$
\sum_{i_0+i_1+\cdots+i_{k-1}=s-k} p_{i_0} p_{i_1} \cdots p_{i_{k-1}} \approx \frac{C_{s-1}^{s-k}}{2^s} \tag{5.40}
$$

由于控制序列 $\underline{s}$ 的周期一般较大(为 2^n-1)，下面近似地把 $\underline{s}$ 视作一个平衡的随机序列。

设 $S_n = \sum_{i=0}^{n-1} s_i$，由概率论中的中心极限定理可知，

$$
\frac{S_n - \frac{n}{2}}{\sqrt{\frac{n}{4}}} \backsim N(0,1)
$$

其中 $N(0,1)$表示正态分布。

设 i_n 表示 $\underline{s}$ 中第 $n+1$ 个非零项(即为 1)，则有

$$
E_n = a_{i_n}, \quad n = 0,1,2,\cdots
$$

因为 $\sum_{i=0}^{i_n} s_i - 1 = n$，所以 $E_{\sum_{i=0}^{i_n} s_i - 1} = a_{i_n}$。由此可知，对任意概率 p，令 $\alpha=1-p$，必存在 u_α，使得若 a_n 在 $\underline{E}$ 中出现，则必以概率 p 落入区间 $\left[\frac{n}{2}-u_\alpha\sqrt{\frac{n}{2}}, \frac{n}{4}+u_\alpha\sqrt{\frac{n}{4}}\right]$之中，将此区间记为 $I_{\frac{n}{2}}$，这里称 a_n 落入 $I_{\frac{n}{2}}$之中意指与 a_n 相对应的 $\underline{E}$ 中元素的角标落入 $I_{\frac{n}{2}}$之中(下同)。显然，$I_{\frac{n}{2}}$中的离散整数是有限的。

5.6.3　拟合序列的构造及符合率的估计

本小节根据上小节的初步理论统计分析结果用择多法构造输入序列 $\underline{a}$ 的拟合序列。

设 $I_{\frac{n}{2}}$中共落入 $2d$ 个 $\underline{a}$ 中的元素，若等于 1 的元素个数不小于 d，令 $a_n'=1$，否则令 $a_n'=0$。这样就得到了 $\underline{a}$ 的拟合序列$\underline{a}'=(a_0',a_1',\cdots)$。

下面定理 5.6.1 给出了符合率的估计。

定理 5.6.1 对收缩序列生成器，设由择多法构造的输入序列 $\underline{a}$ 的拟合序列是 $\underline{a'}$，则有

$$p(a'_n = a_n) = \frac{1}{2} p_n \max\{p_0^{(n)*}, p_1^{(n)*}\} + \frac{1}{2}\left(1 - \frac{1}{2} p_n\right)$$

这里 $p_n = p(\{a_n \text{ 落入 } I_{\frac{n}{2}} \text{之中}\})$，$p_0^{(n)*}$ 和 $p_1^{(n)*}$ 分别表示落入 $I_{\frac{n}{2}}$ 中的 $\underline{a}$ 的元素中 0、1 出现的概率。

证明：$p(a'_n = a_n) = p(a'_n = a_n | a_n \text{ 落入 } \underline{E} \text{ 中}) p(a_n \text{ 落入 } \underline{E} \text{ 中}) + p(a'_n = a_n | a_n \text{ 未落入 } \underline{E} \text{ 中}) p(a_n \text{ 未落入 } \underline{E} \text{ 中}) = \frac{1}{2} p(a'_n = a_n | a_n \text{ 落入 } \underline{E} \text{ 中}) + \frac{1}{4} = \frac{1}{4} + \frac{1}{2} p(a'_n = a_n | a_n \text{ 落入 } I_{\frac{n}{2}}) p(a_n \text{ 落入 } I_{\frac{n}{2}} | a_n \text{ 落入 } \underline{E}) + \frac{1}{2} p(a'_n = a_n | a_n \text{ 落入 } \underline{E} \text{ 但未落入 } I_{\frac{n}{2}}) p(a_n \text{ 未落入 } I_{\frac{n}{2}} | a_n \text{ 落入 } \underline{E}) = \frac{1}{4} + \frac{1}{2} \max\{p_0^{(n)*}, p_1^{(n)*}\} \cdot p_n + \frac{1}{4}(1 - p_n)$。

由证明过程可知，$p(a'_n = a_n) = \frac{1}{2} + \frac{1}{2}(p_n \max\{p_0^{(n)*}, p_1^{(n)*}\} - \frac{1}{2} p_n) \triangleq \frac{1}{2} + \frac{1}{2}\rho_n$。

在实际分析中，人们更关心的是 ρ_n 的估计。显然，ρ_n 的期望值为

$$E\rho_n = p_n E \max\{p_0^{(n)*}, p_1^{(n)*}\} - \frac{1}{2} p_n$$

由于

$$p\left(\max\{p_0^{(n)*}, p_1^{(n)*}\} = \frac{d+i}{2d}\right) = 2C_{2d}^{i+d} \frac{1}{2^{2d}}, \quad 1 \leqslant i \leqslant d$$

$$p\left(\max\{p_0^{(n)*}, p_1^{(n)*}\} = \frac{d}{2d} = \frac{1}{2}\right) = C_{2d}^{d} \frac{1}{2^{2d}}$$

因此，

$$E\max\{p_0^{(n)*}, p_1^{(n)*}\} = 2\sum_{i=1}^{d} \frac{d+i}{2d} C_{2d}^{d+i} \cdot \frac{1}{2^{2d}} + \frac{1}{2} C_{2d}^{d} \frac{1}{2^{2d}}$$
$$= 2\sum_{i=1}^{d} C_{2d}^{d+i-1} \cdot \frac{1}{2^{2d}} + \frac{1}{2} C_{2d}^{d} \frac{1}{2^{2d}} = \frac{1}{2} + \frac{1}{2} \frac{1}{\sqrt{\pi d}}。$$

这里假定了 $\underline{E}$ 中的各比特独立同分布，下同。上式最后一步使用了 Stirling 公式：

$$n! \approx \sqrt{2\pi}\, \mathrm{e}^{-n} n^{n+\frac{1}{2}}$$

下面几处都使用了该公式。

最后得到了 ρ_n 的期望值的估计：

$$E\rho_n = \frac{1}{2} p_n + \frac{1}{2} p_n \cdot \frac{1}{\sqrt{\pi d}} - \frac{1}{2} p_n = \frac{p_n}{2\sqrt{\pi d}}$$

若取 $p_n = 95\%$，则对应的 $d = 1.96\sqrt{\frac{n}{4}}$，此时 $E\rho_n = \frac{0.27}{n^{-\frac{1}{4}}}$。进而还可以求出 ρ_n 的方差

$$D\rho_n = E\rho_n^2 - (E\rho_n)^2$$
$$= \frac{(2d+1)p_n^2}{8d} + \frac{2d-1}{8d} \cdot \frac{p_n^2}{\sqrt{\pi(d-1)}} + \frac{p_n^2}{4\sqrt{\pi d}} - \frac{p_n^2}{4\pi d}$$

5.7 注记

概率论方法与技术最初是从赌博游戏中提出的，但它的应用几乎涉及人们生活和科学技术的各个方面。信息安全研究中自然也少不了概率论方法与技术。例如，在流密码的研究中，人们注意到作为密钥流的序列应该满足一些性质，其中序列的伪随机性是最早被 Golomb[6] 提出并一直是最重要的度量指标之一。Shannon 在保密系统的研究中引入完善保密性的概念，其实质是破译者从密文中得不到关于明文的任何信息，换句话说，就是截获的密文对破译者猜测到原始明文的成功概率没有任何增加。在现代密码学的研究中，Goldwasser 等人又先后提出了概率加密的概念[7] 和概率可检证明系统的概念[8]。概率论方法与技术在密码分析中的应用可参见文献[9]。

参考文献

[1] 威廉・费勒著. 胡迪鹤译. 概率论及其应用. 北京：人民邮电出版社，2006

[2] A. H. 施利亚耶夫著. 周概容译. 概率. 北京：高等教育出版社，2007

[3] Oded Goldreich. Foundations of Cryptography：Basic tools. Publishing House of Electronics Industry，2003. 1

[4] Oded Goldreich. Randomized Methods in Computation. http://www.wisdom.weizmann.ac.il/～oded/rnd.html

[5] 赖瑟著. 李乔译. 组合数学. 北京：科学出版社，1983

[6] Golomb. S W Shift Register Sequences. San Francisco：Holden-Day，1967

[7] Goldwasser S，Micali S. Probabilistic encryption. Journal of Computer and System Sciences.，28(2)：pp. 270-299，1984

[8] Mihir Bellare，Shafi Goldwasser，Carsten Lund，Russeli A. Efficient probabilistically checkable proofs and applications to approximations. STOC 1993：pp. 294-304

[9] 冯登国著. 密码分析学. 北京：清华大学出版社，2000

第6章　计算复杂性方法与技术

无论在理论研究还是在实际应用中，人们经常会遇到各种各样的计算问题，如何度量这些问题的难易程度，在计算机科学中具有十分重要的意义。计算复杂性理论就是针对求解问题所需要的计算时间和空间等资源进行研究，给出求解一个问题是“容易的”还是“困难的”确切定义，并依据求解计算问题所需要的时间和空间对问题进行分类。

计算复杂性理论是设计和分析密码算法与安全协议的基础，很多密码算法与安全协议的安全性是以某些计算问题的困难性为前提条件的。例如，在密码算法的设计与分析研究中，使用多项式归约技术的可证明安全性理论得到广泛应用，即通过有效的归约转化，将对密码算法的任何有效的攻击归约到求解一类已知的 $\mathcal{NP}$ 难问题的一个实例。通常，该类 $\mathcal{NP}$ 难问题是广泛接受的困难问题，对于密码算法的安全性，这样的证明方式提供了很高的可信度。

本章主要介绍了信息安全研究中常用的计算复杂性理论的基本概念、基本原理、典型的归约方法和模型，并介绍了计算复杂性方法与技术在密码算法和安全协议设计与分析中的应用。

6.1　基本概念

说一个问题是可解的，是指能够编制一个计算机程序，只要运行足够长的时间，使用足够多的空间，该程序对任何输入都能给出正确的回答。可以求解某一问题并一步一步地执行的计算过程称为算法。在20世纪30年代，许多数学家都曾致力于问题的可解性研究，如歌德尔、图灵和丘奇等，这些研究工作表明，有许多问题都是不可解的。

一个问题在理论上是可解的，并不意味着该问题在实际中也是可解的。对于许多问题来说，发现求解该问题的一个算法是容易的，但是利用该算法得到该问题的解在实际中是不可行的。著名的旅行商问题就是一个例子。旅行商问题简单描述如下：一个推销商从当前所在的城市出发，到其他 $n-1$ 个城市去推销产品，然后返回其所在的城市。假定任意两个城市之间都有一条线路，如何选择一条周游路线，使得旅行商经过每个城市一次且仅一次，并且使得他所走过的路径最短？求解该问题的一个简单算法就是穷举所有的周游路线，从中挑出最短的一条。一共有 $(n-1)!/2$ 条不同的周游路线，求一条周游路线的长度需要 $n-1$ 次加法运算，所以这个算法共需要 $(n-1)\cdot(n-1)!/2$ 次加法运算。当 $n=50$ 时，该算法需要 $49\times 49!/2\approx 1.5\times 10^{64}$ 次加法运算。假如一台计算机每秒钟执行 10^8 次加法运算，那么该算法大约需要执行 10^{49} 年，这是一个不可接受的计算时间。所以，当 n 较大时，该算法在实际中是不可能求解旅行商问题的。

这个例子说明,可计算问题在理论与实际中是存在差别的。所有的计算机都需要占用一定的资源,可以用执行该算法所需要的运行时间和内存空间来度量。一个算法所需要占用的资源的数量,就是评估其实际可行性的自然途径。简单地说,如果一个问题可以在"合理的"时间内,使用一个"不太大"的计算机求解,那么该问题自然就是可解的。求解问题的算法对时间资源的需求具有实际的重要意义,算法所花费的时间越少,则该算法越好也越"有效"。同样,算法对空间的需求也具有重要的实际意义。算法执行所花费的时间和空间的分析已经成为计算机科学中的一个重要研究内容。

计算复杂性理论,就是从求解问题的实际困难出发,在理论上对计算机可解的问题进行分类。为了说明一个问题是"容易的",只需要给出一个实际的求解算法即可。但是,要说明一个问题本质上是"困难的",需要证明实际的求解算法是不存在的。事实上这是非常困难的。

给定一个问题 P,其复杂性是由求解 P 的算法的时间复杂性来确定的。如果存在一个有效的算法来求解 P,则说问题 P 是容易的;如果不存在这样的有效算法,则称 P 是困难的。因此,为了弄清楚问题的复杂性,首先需要给出算法有效性的确切定义。

6.1.1 图灵机

一个算法的运行时间受到多种因素的影响,如选用的计算语言、编写程序的方法及计算所用的计算机等。为了克服这一困难,精确地定义有效算法的概念,在计算复杂性理论研究中,采用了统一的计算模型。这一模型是英国数学家图灵(A. Turing)于 1936 年提出的,后人称之为图灵机。图灵机计算模型后来被证明是一种非常通用的计算模型。下面介绍图灵机的一个变型,用于理解计算复杂性。

图灵机由有效状态单元、k 条纸带以及同样数量的读写头组成。有限控制单元控制磁头读、写纸带的操作,每个读写头访问一条纸带,沿着纸带向左或者向右移动完成这一操作。每一条纸带分成无限个单元。图灵机求解一个问题时,读写头扫描一个有限字符串,从纸带最左边的单元开始按照顺序存放在纸带上,每个字符占用一个单元,其右边剩下的是空白单元,该字符串称为问题的一个输入。扫描过程从含有输入的纸带左端开始,同时图灵机赋予一个初态。任何时刻图灵机都只有一个读写头访问其纸带。读写头对纸带的一次访问称为一个移动。如果图灵机从初始状态开始,一步一步地合法移动,完成对输入串的扫描,最终满足中止条件而停下来,则称图灵机识别了该输入;否则,图灵机在某一点没有合法移动,它会没有识别输入就停下来。图灵机识别的输入称为可识别语言的一个实例。

为了识别一个输入,图灵机 M 在停下来之前所移动的步数称为 M 的运行时间或者 M 的时间复杂度,记为 T_M。很明显,T_M 可以表示为函数 $T_M(n): N \to N$,其中 n 是输入实例的长度或者规模,也就是说,当 M 在初始状态时,它就是组成输入串的字符数。显然 $T_M(n) \geqslant n$。除了对时间的要求外,M 还有空间的要求,即 M 在操作中读写头访问的纸带单元数,它可以表示为函数 $S_M(n): N \to N$,称为 M 的空间复杂度。

6.1.2 算法的表示

图灵机提供了一种通用的计算模型，给出了度量程序计算复杂性的一个准确概念。但在实际中，一般并不希望按照这种原型机器来描述算法，甚至不希望按照现代计算机的微指令来描述。为了清楚、有效地描述算法和数学命题，使用一种高级编程语言，称为“准编程语言”，它和一些通用的高级编程语言很接近，诸如 Pascal 或者 C。由于它具有不言自明的清晰特征，理解起来不会有任何困难。

实际上在第 1、2 章中已经使用过这种表示方法，这里再通过一个具体例子来说明一下。考察 1.3.1 小节介绍的欧氏算法，这是计算两个整数的最大公因子的著名算法。以 $\gcd(a,b)$表示整数 a 和 b 的最大公因子，即整除 a 和 b 的最大整数。

可将 1.3.1 小节介绍的欧氏算法表示成算法 6.1.1 和算法 6.1.2，分别称之为欧氏算法和扩展欧氏算法。

算法 6.1.1 欧氏算法。

输入 整数 $a>b\geqslant 0$；

输出 $\gcd(a,b)$。

1. If $b=0$ return(a)；
2. Return($\gcd(b, a \bmod b)$)。

算法 6.1.2 扩展欧氏算法。

输入 整数 a 和 b，满足 $a>b\geqslant 0$；

输出 整数 s 和 t，满足 $as+bt=\gcd(a,b)$。

1. $i\leftarrow 0; r_{-1}\leftarrow a; r_0\leftarrow b$；
   $s_{-1}\leftarrow 1; t_{-1}\leftarrow 0; s_0\leftarrow 0; t_0\leftarrow 1$
2. while $r_i=as_i+bt_i\neq 0$, do
   (a) $q\leftarrow r_{i-1}\div r_i$；
   (b) $s_{i+1}\leftarrow s_{i-1}-qs_i; t_{i+1}\leftarrow t_{i-1}-qt_i$
   (c) $i\leftarrow i+1$
3. return $((s_{i-1}, t_{i-1}))$。

显然，算法 6.1.1 和算法 6.1.2 的递归调用次数等于算法 6.1.2 的循环次数，即 k。

考虑 $a>b$ 的情形，其中$|x|$表示 x 的二进制的长度。对于 $i=1,2,\cdots,k-1$，有$|r_{i+1}|<|r_{i-1}|$，所以 k 的最大值以 $2|a|$ 为界。如果将模运算视为基本运算，那么使用算法 6.1.1 实现 gcd 的时间复杂度的界为 $2|a|$，即关于 a 的规模的线性函数。因此，有以下结论：

计算 a 和 b 的最大公因子 $\gcd(a,b)$至多需要执行 $2\max(|a|,|b|)$次模运算。即算法 6.1.1 和算法 6.1.2 将在 $2\max(|a|,|b|)$次循环内终止。

6.1.3 计算复杂度的表示方法

当衡量一个算法的计算复杂度时，通常很难（也没有必要）确切地给出复杂性度

量表示式中的常数。实际上，由于算法的计算复杂度是其输入的规模 n 的函数，要比较两个算法的计算复杂度，只需要考虑当 n 充分大时它们随着 n 的增大而增大的量级即可。为了简化计算复杂度的度量任务，需要引入"阶"的概念。

定义 6.1.1 设函数 $f(n)$ 和 $g(n)$ 为两个正整数函数，如果存在常数 $c>0$ 和正整数 N，使得当 $n>N$ 时有 $g(n)<c|f(n)|$，则称 $g(n)=O(f(n))$。如果 $g(n)=O(f(n))$，并且 $f(n)=O(g(n))$，则称 $f(n)=\Theta(g(n))$。

不难证明，O 和 Θ 具有以下性质。

(1) 如果 $f(n)=O(g(n))$，$g(n)=O(h(n))$，则 $f(n)=O(h(n))$。

如果 $f(n)=\Theta(g(n))$，$g(n)=\Theta(h(n))$，则 $f(n)=\Theta(h(n))$。

(2) 如果 $f(n)$ 是一个 d 次多项式，那么 $f(n)=\Theta(n^d)$，$f(n)=O(n^{d'})(d'\geqslant d)$。

(3) 对于任意多项式 $p(n)$ 以及任意整数 $m>1$，有 $p(n)=O(m^n)$。

使用记号 $O(\cdot)$ 可以将算法 6.1.1 和算法 6.1.2 的计算复杂度表示为 $O(\log a)$。注意在这个表达式中，使用 $\log a$ 代替了 $|a|$，而没有明确说明对数的底数。容易验证，对于任意的底数，它都给出了正确的复杂性度量表示。

到目前为止，一直把一次模运算的计算时间代价看作一个单位时间，即其时间复杂度是 $O(1)$。实际上，模运算包含了除法，其时间复杂度和除法的时间复杂度基本上是一样的。从这个角度来看，用 $O(1)$ 表示除法的时间复杂度有些太粗略了。

下面通过按比特计算来度量算术运算。在按比特计算中，所有变量的取值要么是 0，要么是 1，而运算是逻辑运算，即与、或、异或和非。

在按比特运算模式下，两个整数 i 和 j 之间的加法和减法需要 $\max\{\log i,\log j\}$ 次运算，即时间复杂度是 $O(\max\{\log i,\log j\})$，而两个整数 i 和 j 之间的乘法和除法需要时间为 $O(\log i\cdot\log j)$。当然，乘法和除法也存在更小的时间复杂度，如通过离散傅里叶变换可以得到 $O(\log(i+j)\cdot\log\log(i+j))$，但 O 项的常系数比较大。

现在精确度量算法 6.1.1 和算法 6.1.2 的时间复杂度。根据 1.3.1 小节中的讨论可知，对于 $a>b$，$\gcd(a,b)$ 可以在时间 $O(\log a)$ 内计算出来。考虑到，模运算和除法的时间代价是 $O((\log a)^2)$，直观上算法 6.1.1 和算法 6.1.2 的时间复杂度是 $O((\log a)^3)$。

事实上，欧氏算法有更好的估计。注意到，计算除法 $a=bq+r$ 的时间复杂代价为 $O((\log a)(\log q))$。对于欧氏算法的中间商 $q_1,q_2,\cdots,q_k$，有

$$\sum_{i=1}^{k}\log q_i=\log\left(\prod_{i=1}^{k}q_i\right)\leqslant\log a$$

因此，计算最大公因子总的时间代价不超过

$$\left(\sum_{i=1}^{k}O((\log a)(\log q_i))\leqslant O((\log a)^2\right)$$

对于一个图灵机 M 来说，如果存在正整数 d，使得 $T_M(n)=\Theta(n^d)$，则称 M 为多项式时间算法。如果 $T_M(n)=\Theta(2^{\log n})$，则称 M 为亚指数时间算法。如果 $T_M(n)=\Theta(2^n)$，则称 M 为指数时间算法。

6.2 基本原理

一个问题的计算复杂度是由解决这个问题的算法的计算复杂度来决定的。由于解决一个问题的算法可能有多个,其计算复杂度也各不相同,所以,在理论上定义一个问题的计算复杂度为求解该问题的最有效算法的计算复杂度。实际上,要证明一个算法是求解某一问题的最有效的算法是很困难的,所以只能把求解问题的计算复杂度粗略地分为 3 类,即 $\mathcal{P}$类(确定性多项式时间)、$\mathcal{NP}$类(非确定性多项式时间)和$\mathcal{NPC}$类(NP 完全类)。

6.2.1 多项式时间可识别语言

设函数 $p(n)$是整数上关于 n 的一个多项式,具有形式 $p(n)=c_kn^k+c_{k-1}n^{k-1}+\cdots+c_1n+c_0$,其中,$k$ 和 $c_i(i=0,1,\cdots,k)$是整数。

定义 6.2.1 用 $\mathcal{P}$表示具有以下特征的语言类:对于语言 L,如果存在一个图灵机 M 和一个多项式 $p(n)$使得对任意的非负整数 n,M 可以在时间 $T_M(n)\leqslant p(n)$内识别任意实例 $I\in L$,则称语言 L 在 $\mathcal{P}$中,其中整数参数 n 表示实例 I 的规模,并称 L 是可以在多项式时间内识别的语言。

粗略地讲,可以在多项式时间内识别的语言总是很“容易的”,换句话说,多项式时间图灵机被认为总是很“有效的”。识别 $\mathcal{P}$中语言的图灵机都是确定性的,其输出结果完全取决于输入和初始状态。也就是说,对同样的输入和初始状态,两次运行一个确定性图灵机,得到的输出结果是相同的。

例 6.2.1(DIV3 语言) 设 Div3 表示被 3 整除的非负整数集,证明 Div3$\in\mathcal{P}$。

构造一个多项式时间内识别 Div3 的单向图灵机来证明。首先注意到,如果将输入写成一个三进制表示的整数,也就是说,输入是{0,1,2}中的字符串,那么识别该问题就变得非常简单:输入 x 属于 Div3 当且仅当 x 的最后一位是 0。因此,构造的图灵机只需要向右端移动直到一个空白字符,然后停下来,当且仅当最后一个非空的字符是 0,回答“是”。显然,这个图灵机可以在移动实例规模步数内识别该实例。因此,Div3$\in\mathcal{P}$。

不过,希望证明 Div3$\in\mathcal{P}$与输入的表示方式无关。这里只需证明输入表示为二进制的情形。设这样的一个图灵机称为 Div3。Div3 的有限状态控制按照表 6.1 所示一步步移动。

表 6.1 Div3 图灵机示意图

当前状态	纸带上的符号	下一步移动	下一个状态
q_0(初态)	0	右	q_0
	1	右	q_0
	空白	“响铃”或终止	q_1
q_1	0	右	q_2
	1	右	q_0
q_2	0	右	q_1
	1	右	q_2

下面说明表 6.1 所定义的图灵机 Div3 可以识别 Div3 中的所有实例。

注意到，要识别一个二进制串 $x \in$ Div3 是否成立，Div3 只需要 3 个状态，分别对应于其(读写头)完成扫描串 $3k$、$3k+1$ 和 $3k+2(k \geqslant 0)$ 的情形。最小输入实例 0 约定 Div3 在完成扫描输入串 0 时必定处于初态(不失一般性，设初态为 q_0)。在完成扫描输入串 1 时，可以指定 Div3 为状态 q_1，完成扫描输入串 2 时，Div3 为状态 q_2。对任何二进制表示的非负整数 a，后面跟一个 0(或 1)得到值 $2a$(或 $2a+1$)。因此，完成对 $a=3k$(当 Div3 初态为 q_0 时)的扫描后，由于在该点完成扫描 $2a=6k=3k'$，当下一步扫描到字符 0 时，Div3 必定仍在状态 q_0；下一步扫描到字符 1 时必定移动到 q_1，因为在该点完成扫描 $2a+1=6k+1=3k'+1$。类似地，完成对 $a=3k+1$(当 Div3 在状态 q_1 时)的扫描后，当完成扫描 $2a=6k+2=3k'+2$ 时，Div3 必定移动到 q_2；当完成扫描 $2a+1=6k+3=3k'$时，必定移动到 q_0。对于 $a=3k+2a=3k+2$ 还有两种情况：$2a=6k+4=3k'+1$(Div3 从 q_2 移动到 q_1)和 $2a+1=6k+5=3k'+2$(Div3 停留在 q_2)。

因此，对于任意的 $k \geqslant 0$，3 个状态分别对应于 Div3 完成扫描 $3k$、$3k+1$ 和 $3k+2$。一旦读写头遇到特殊的字符串“空”，只有在状态 q_0 的 Div3 才设置响铃并停止移动(表示终止并回答“是”)，从而识别出输入 $3k$；对于其他两种状态，Div3 没有合法移动，因此没有识别就终止了。

显然，$T_{\text{Div3}}(n)=n$。因此，Div3 确实可以在多项式时间内识别 Div3 语言。

6.2.2 多项式时间计算问题

根据定义，$\mathcal{P}$是多项式时间可识别语言类。语言识别问题是一个判定性问题。对任意的输入，一个判定性问题的输出为“是”或“否”。但是 $\mathcal{P}$类是非常普遍的，包括多项式时间的计算性问题。对任意可能的输入，计算性问题要求输出比“是”或“否”更一般的答案。既然图灵机可以向纸带写入字符，当然能够输出比“是”或“否”更一般的答案。

举例来说，可以设计一种图灵机，它不仅能够识别任意实例 $x \in$ Div3，而且在识别后还能输出 $x/3$。把这个新的图灵机称为 Div3-Comp。实现 Div3-Comp 的一个非常简单的方法是将输出写成三进制表示：输入是 Div3 中的一个实例，当且仅当其最后一位是 0，在识别该输入之后，图灵机的输出就是输入纸带上的内容去掉最后一个 0，除非纸带上全部是 0。如果要求 Div3-Comp 只输入和输出二进制数，那么 Div3-Comp 可按照下列方式实现：首先将输入 x 从二进制转换为三进制，一旦获得了三进制表示的 $x/3$，再转换为二进制表示作为输出。显然，转换可以机械地逐位进行，需要 $C|x|$ 次移动，其中 C 是一个常数。

从这个例子可以看出：$\mathcal{P}$类必定包括 Div3-Comp 可以解决的问题。实际上，$\mathcal{P}$包括所有的多项式时间计算问题。

6.2.3 概率多项式时间可识别语言

如果一个语言不属于 $\mathcal{P}$，那么不存在总能有效地识别它的图灵机。但是，有一类

语言具有以下特性：没有证明它们属于 $\mathcal{P}$，但它们总能用一种图灵机有效地识别，尽管有时也会出错误。这种机器有时出错的原因，是在有些操作步骤中机器会做随机移动。有些移动会产生正确的结果，而其他移动则会产生错误的结果。这种图灵机称为非确定性图灵机。

这里，人们关心判定性问题的一个子类，它具有下面的有界差错特征："回答判定性问题时，非确定性图灵机出错概率的界是一个常数(其概率空间取自于随机纸带)"。

习惯上，称具有有界差错概率的非确定性图灵机为概率图灵机。而"非确定性图灵机"则专指另一类不同的判定性问题(见 6.2.5 小节)。概率图灵机实质上是指有随机输入并在其输入的长度的多项式时间一定停机的图灵机。

概率图灵机有多条纸带，其中有一条称为随机纸带，上面是一些均匀分布的随机字符。在扫描一个输入实例 I 时，机器将和随机纸带交互，读取一个随机字符，然后像确定性图灵机一样工作。该随机串称为概率图灵机的随机输入。概率图灵机对输入 I 的识别不再是 I 的确定性函数，而是与一个随机变量(即随机输入)有关的函数。该随机变量对识别 I 造成了一定的差错概率。

人们把概率图灵机可识别的语言类称为概率多项式时间(PPT)语言，用 $\mathcal{PP}$ 表示。

定义 6.2.2 用 $\mathcal{PP}$ 表示具有下列特征的语言类：称一个语言 L 属于 $\mathcal{PP}$，如果存在一个概率图灵机 P_M 和一个多项式 $p(n)$，对任意的非负整数 n，P_M 可以在时间 $T_{PM}(n)\leqslant p(n)$ 内以一定的差错概率识别任意实例 $I\in L$，其中差错概率是关于 P_M 随机移动的一个随机变量，整数参数 n 表示实例 I 的规模。

在定义 6.2.2 中，"一定的差错概率"可以表示为以下两个条件概率界：

$$\Pr[P_M \text{ 识别 } I \in L \mid I \in L] \geqslant \varepsilon$$

$$\Pr[P_M \text{ 识别 } I \in L \mid I \notin L] \leqslant \delta$$

其中，ε 和 δ 是参数，满足

$$\varepsilon \in (0.5, 1], \quad \delta \in [0, 0.5)$$

这里概率空间是 P_M 的随机纸带。

ε 是正确识别一个实例的概率界，称为完备性概率界。其等价形式可表示为

$$\Pr[P_M \text{ 识别 } I \notin L \mid I \in L] < 1-\varepsilon$$

在该式中，$1-\varepsilon$ 表示错误地拒绝的概率界。

δ 是错误识别一个非实例的概率界，称为可靠性概率界。

对于概率多项式时间图灵机 P_M，对输入 I 重复运行 n 次 P_M，表示为 $P'_M(I,n)$，它仍然表示一个概率多项式图灵机。可以通过"大数判别"作为 $P'_M(I,n)$ 接受或者拒绝 I 的准则，也就是说，如果 $\lfloor n/2 \rfloor+1$ 或者更多次 $P_M(I)$ 运行都输出接受(或拒绝)，那么 $P'_M(I,n)$ 就接受(或拒绝)。显然，$P'_M(I,n)$ 的可靠性和完备性概率是关于 n 的函数。实际上，$P'_M(I,n)$ 仍然是关于 I 的规模的多项式时间。

由于 n 次运行 $P_M(I)$ 的随机移动是相互独立的，失败的概率为 $1-\varepsilon$(或者 $1-\delta$)，根据二项分布，大数判别准则给出 $P'_M(I,n)$ 的差错概率界是 n 次伯努利实验成功

了$\lfloor n/2 \rfloor+1$ 次或更多次的概率之和。对于完备性，有

$$\varepsilon(n) = \Pr[\xi_n \geqslant \lfloor n/2 \rfloor + 1] = \sum_{i=\lfloor n/2 \rfloor+1}^{n} b(i;n,\varepsilon)$$

对于可靠性，有

$$\delta(n) = \Pr[\eta_n \geqslant \lfloor n/2 \rfloor + 1] = \sum_{j=\lfloor n/2 \rfloor+1}^{n} b(j;n,\delta)$$

这两个表达式都是各自二项分布的累加。因为 $\delta<0.5<\varepsilon$，第一个分布的中心项在$(n+1)\varepsilon>\lfloor n/2 \rfloor+1$ 点（二项式的项在该点达到最大值）；另一个分布的中心项在$(n+1)\delta<\lfloor n/2 \rfloor+1$ 点。推导可得

$$\delta(n) < \frac{2(1-\delta)}{(1-2\delta)^2} \cdot \frac{1}{n+1}$$

由于 δ 是常数，则有

$$\delta(n) \to 0 (n \to \infty)$$

类似地，

$$\varepsilon(n) > 1 - \frac{c}{n}$$

其中 c 是常数。

令 $n=|I|$，则图灵机 $P'_M(I,n)$的运行时间为$|I| \cdot \mathrm{poly}(|I|)$，其中 $\mathrm{poly}(|I|)$是 P_M 对实例 I 的运行时间。因此，P'_M仍然是多项式时间的。

下面用大数定律或概率不等式给出在密码学中针对 $\mathcal{PP}$ 语言类经常用到的一个放大/缩小概率的界的基本技巧。

定理 6.2.1　$L\in\mathcal{PP}$ 当且仅当存在概率图灵机 P_M，使得

1）$\Pr[P_M$ 识别 $I\in L \mid I\in L]\geqslant\frac{2}{3}$，

$\Pr[P_M$ 不能识别 $I\in L \mid I\notin L]\geqslant\frac{2}{3}$。

当且仅当存在多项式 $p(\cdot)$和概率图灵机 P_M，使得

2）$\Pr[P_M$ 识别 $I\in L \mid I\in L]\geqslant\frac{1}{2}+\frac{1}{p(|I|)}$，

$\Pr[P_M$ 不能识别 $I\in L \mid I\notin L]\geqslant\frac{1}{2}+\frac{1}{p(|I|)}$。

当且仅当存在多项式 $p(\cdot)$和概率图灵机 P_M，使得

3）$\Pr[P_M$ 识别 $I\in L \mid I\in L]\geqslant 1-2^{-p(|I|)}$，

$\Pr[P_M$ 不能识别 $I\in L \mid I\notin L]\geqslant 1-2^{-p(|I|)}$。

只证明 1）和 2）等价，借助更强的概率不等式（大数定律），如 Chernoff 不等式，可类似地证明 1）和 3）等价。

证明：条件 1）成立，显然条件 2）成立。下证条件 2）成立，条件 1）亦成立。

构造机器 P'_M如下：对输入 I，P'_M调用机器 P_M，输入 I，运行 $O(p(|I|)^2)$次，P'_M的输出为超过半数的 P_M 的输出。

对给定的 I，当 $M(I)$可识别 I 时，令 $M(I)=1$，当 $M(I)$不能识别 I 时，令

$M(I)=0$，因此可将 $M(I)$ 视作取值为 0、1 的随机变量。令 $n=p(|I|)^2$，$M_k=M(I)$，$k=1,\cdots,n$，就得到 n 个独立的同分布的随机变量。设 p' 是 $M(I)=1$ 的概率。由假设，对 $I\in L$，$p'\geqslant\frac{1}{2}+\frac{1}{p(|I|)}$。由 M' 的定义，有

$$\begin{aligned}\Pr[M'(I)=0]&=\Pr\left[\frac{1}{n}\sum_{k=1}^{n}M_k\leqslant\frac{1}{2}\right]=\Pr\left[\frac{1}{n}\sum_{k=1}^{n}M_k-p'\leqslant\frac{1}{2}-p'\right]\\&\leqslant\Pr\left[\frac{1}{n}\sum_{k=1}^{n}M_k-p'\leqslant-\frac{1}{p(|I|)}\right]\\&\leqslant\Pr\left[\left|\frac{1}{n}\sum_{k=1}^{n}M_k-p'\right|\geqslant\frac{1}{p(|I|)}\right]\\&\leqslant\frac{p'(1-p')}{p(|I|)^{-2}p(|I|)^2}\\&=p'(1-p')\leqslant\frac{1}{4}\end{aligned}$$

其中，倒数第二个不等式是根据定理 5.5.3 得到的。
所以

$$\Pr[M'(I)=1]=1-\Pr[M'(I)=0]\geqslant\frac{3}{4}>\frac{2}{3}$$

对 $I\notin L$，有 $p'\leqslant\frac{1}{2}-\frac{1}{p(|I|)}$。于是

$$\begin{aligned}\Pr[M'(I)=1]&=\Pr\left[\frac{1}{n}\sum_{k=1}^{n}M_k\geqslant\frac{1}{2}\right]=\Pr\left[\frac{1}{n}\sum_{k=1}^{n}M_k-p'\geqslant\frac{1}{2}-p'\right]\\&\leqslant\Pr\left[\frac{1}{n}\sum_{k=1}^{n}M_k-p'\geqslant\frac{1}{p(|I|)}\right]\\&\leqslant\Pr\left[\left|\frac{1}{n}\sum_{k=1}^{n}M_k-p'\right|\geqslant\frac{1}{p(|I|)}\right]\\&\leqslant\frac{p'(1-p')}{p(|I|)^{-2}p(|I|)^2}\\&=p'(1-p')\leqslant\frac{1}{4}\end{aligned}$$

从而

$$\Pr[M'(I)=0]=1-\Pr[M'(I)=1]\geqslant\frac{3}{4}>\frac{2}{3}$$

例 6.2.2 全体素数的集合(记为 PRIMES)属于 $\mathcal{PP}$ 类。

我们可根据算法 1.3.2(记为 Prime_Test(.))检测一个奇数 p 是否是一个素数。

根据费马小定理，如果 p 是素数，那么 Prime_Test(p)总是返回"p 是素数"，即

$$\Pr[x^{(p-1)/2}\equiv\pm1(\mathrm{mod}\ p)\mid p\text{ 是素数}]=1$$

另一方面，如果 p 是一个合数，那么同余式 $x^{(p-1)/2}\equiv\pm1(\mathrm{mod}\ p)$ 一般不会成立。事实上，对于 $x\in(1,p-1]$，$\gcd(x,p)=1$，有

$$\Pr[x^{(p-1)/2} \equiv \pm 1 (\bmod\ p) \mid p \text{ 是合数}] \leqslant 1/2$$

因此，如果均匀随机选取 x 通过了 k 次检测（当然 -1 的情形至少出现一次），那么 p 不是素数的概率不大于 2^{-k}。这里，使用了“一致性判别准则”：在 $\log_2 p$ 次检测中只要有一次失败，这个 p 就被拒绝。注意这一判别准则与大数判别准则不同，后者可以容忍失败，只要失败的次数不超过半数。一致性判别的可靠性概率趋近于 0 的速度比大数判别的情形快得多。

令 $k=\log_2 p$，对任何输入实例 p，有

$$\Pr[\text{Prime_Test}(p) = \text{“}p \text{ 是素数”} \mid p \text{ 不是素数}] \leqslant 2^{-\log_2 p}$$

注意到，对于长度为 $\log_2 p$ 比特的输入，计算模指数和最大公因子的时间复杂度都是 $O((\log_2 p)^3)$，因此，Prime_Test(p)的时间复杂度的界为 $O((\log_2 p)^4)$。

需要说明的是，2002 年 8 月，3 位印度计算机科学家 Agrawal、Kayal 和 Saena 发现了一种确定性多项式时间的素性检测算法，因此，PRIMES 事实上属于 $\mathcal{P}$。

6.2.4　有效算法

能够解决多项式时间可识别语言类或者概率多项式时间可识别语言类中的问题的算法称为有效算法。

定义 6.2.3　对于一个算法，无论是确定性或者是随机化的，如果其运行时间可以表示为输入规模的多项式，则称该算法是有效算法。

下面通过一个例子来说明有效算法。

概率素性检测的思想可以直接转化为一个算法，该算法能够生成给定长度的随机概率素数。算法 6.2.1 给出了具体步骤。

算法 6.2.1　随机 k 比特概率素数生成。

输入　一个正整数

输出　一个 k 比特的随机素数

Prime_Gen(k)

1. $p \in (2^{k-1}, 2^k - 1]$，p 为奇数；
2. 如果 Prime_Test(p)=“p 不是素数”，return(Prime_Gen(k))；
3. Return(p)。

首先假设 Prime_Gen(k)会终止，那么该算法最终找到了一个数 p，满足 Prime_Test(p)=“p 是素数”。根据对 Prime_Test 的差错概率界估计，输出的 p 不是素数的概率上界为 2^{-k}，其中 $k=\log_2 p$。

显然，有这样一个问题：Prime_Gen(k)最终能够停下来吗？

著名的素数定理表明，小于 X 的素数大约有 $X/\log X$ 个。因此 k 比特的素数的个数大约为

$$\frac{2^k}{k} - \frac{2^{k-1}}{k-1} \approx \frac{2^k}{2k}$$

因此，可以期望 Prime_Gen(k)在第二步递归调用自身 $2k$ 次才可以找到一个概率素

数，最终停止下来。

由于 Prime_Test(p)的时间复杂度为 $O((\log p)^4)=O(k^4)$，调用 $2k$ 次 Prime_Test 后，Prime_Gen(k)的时间复杂度为 $O(k^5)$。

有效算法这一定义给出了容易处理问题的概念：不管是确定性还是随机化的，多项式时间的问题是容易处理的，也就是说，即使这类问题的输入规模非常大，它要求的资源也是可以处理的。相对地，易处理之外的问题就是难处理的。

有许多函数大于任意的多项式。设函数 $f(n): N\rightarrow R$，如果对于任意的多项式 $p(n)$，存在自然数 n_0，使得对任意的 $n>n_0$，都有 $f(n)>p(n)$，那么就说 $f(n)$是关于 n 的任意多项式无界的。

例如，对于任意的 $a>1, 0<t<1$，函数 $f_1(n)=a^{n^t(\log n)^{1-t}}$，$f_2(n)=n^{(\log\log\log n)^t}$ 都是关于 n 的任意多项式无界的。

与多项式时间(确定性的或非确定性的)问题相比较，时间复杂度为非多项式时间界的问题视为在计算上不可行的，或者困难的。这是因为，当问题实例的规模增长时，求解这种问题所要求的资源增长太快，以至于很快就大得不实际了。例如，设 N 是长度为 n 的合数，即 $n=\log N$，那么，取 $a\approx\exp(1.9229994\cdots+o(1))$ 和 $t=1/3$，其中 $o(1)\approx 1/\log N$，函数 $f_1(\log N)$的时间复杂度表达式为

$$\exp(1.9229994\cdots+o(1)(\log N)^{1/3}(\log\log N)^{2/3})$$

这是关于 N 的亚指数表示。如果把 1/3 替换为 1，那么该形式就是指数表示。亚指数函数比指数函数增长缓慢得多，但是比多项式函数快得多。多于 1024 比特的整数来说，上式是一个大于 2^{86} 的量，即使使用大量的计算机并行运行，这样的量级也是无法处理的。

对于很大的量，定义了非多项式界的概念，对于很小的量，定义下面的概念。

设函数 $\varepsilon(n): N\rightarrow R$，如果 $1/\varepsilon(n)$是关于 n 的一个非多项式有界的量，那么就称 $\varepsilon(n)$是关于 n 的可忽略量，简称 $\varepsilon(n)$是可忽略的。

例如，对于任意的多项式 $p(n)$来说，$p(n)/2^n$ 就是一个可忽略量。

可忽略量比任意的多项式的倒数更快地趋近于 0。如果认为非多项式有界量是难处理的，那么，忽略任何可忽略量应该是无关紧要的。

6.2.5 非确定性多项式时间

考虑下面的判定性问题。

问题：SQUARE-FREENERSS。

输入：N：一个正的奇合数。

输出：N 无平方因子吗？

如果不存在素数 p，满足 $p^2 \mid N$，回答 YES。

SQUARE-FREENERSS 问题很困难，到目前为止，还没有已知的算法(无论是确定性算法还是概率算法)可以在多项式时间内回答这一问题。存在一些回答该问题的算法，如下面的例子：输入 N，用所有不超过$\lfloor\sqrt{N}\rfloor$的奇素数的平方穷举试除，如果所有试除都失败，则回答 YES。对一般的输入实例 N，这种方法的运行时间

为 $O(\lfloor\sqrt{N}\rfloor)=O(\mathrm{e}^{0.5\log N})$。

不过，也不应该将 SQUARE-FREENERSS 问题想得太难了。如果知道该问题的一些“内部信息”，称为证据（或者辅助输入），那么回答可以在输入长度的多项式时间内验证。例如，对于输入 N，整数 $\varphi(N)$ 称为 N 的欧拉函数，是小于 N 而且与 N 互素的所有整数的个数，它可以用作一个证据，使一个有效的验证算法可以验证关于 N 是否无平方因子的回答。算法 6.2.2 就是一个有效的验证算法。

算法 6.2.2 Square-Free(N, $\varphi(N)$)

1. $d\leftarrow \gcd(N,\varphi(N))$；
2. 如果 $d=1$ 或者 d^2 不整除 N，回答 YES，否则回答 NO。

根据 $\varphi(N)$ 的定义很容易证明该算法的合理性。从欧氏算法的时间复杂度可知，上述算法的运行时间是关于 N 的长度的多项式时间。

现在描述这样一个计算设备，它给出求解和 SQUARE-FREENERSS 问题具有相同特征的一类问题的方法。该设备的计算可以通过一棵树来描述。

这一设备称为非确定性图灵机。它是图灵机的一个变型，在每一步，机器具有有限个选择进行下一步。一个输入串称为可识别的，条件是至少存在一系列合法的移动，当机器扫描第一个输入符号时，它从机器初始状态开始，完成扫描输入串后到达满足中止条件的状态。这样的一系列移动称为一个识别序列。

可以想象，非确定性图灵机寻求可识别问题实例的解需要进行一系列的猜测，正确猜测的系列移动形成一个识别序列。因此，机器可以做出的所有可能移动形成了一棵树（称为非确定性图灵机的计算树）。树的大小（即节点数）显然是关于输入规模的指数函数。但是，对于可识别的输入实例，由于一个识别序列中移动的次数就是树的深度 d，识别序列中移动的次数必定以输入实例规模的多项式为界。所以，通过一系列正确的猜测，识别可识别的输入的时间复杂度是输入规模的一个多项式。

定义 6.2.4 人们称非确定性图灵机在多项式时间内可以识别的语言类为 $\mathcal{NP}$ 类。

直接可以看出

$$\mathcal{P}\subseteq\mathcal{NP}$$

也就是说，$\mathcal{P}$ 中的每一种语言（判定性问题）用非确定性图灵机是很容易识别的。$\mathcal{NP}$ 问题的这些子类可以有效求解的原因在于这些问题有大量的证据，通过随机猜测很容易找到。以 $\mathcal{NP}$ 难问题表示只有稀疏证据的非确定性多项式时间（判定性）问题，这里稀疏证据的含义是：在一个 $\mathcal{NP}$ 问题的计算树中，识别序列个数相对于序列总数而言是一个可忽略的量。

如果某一个问题只有稀疏证据，那么非确定性图灵机实际上并不能提供任何有效的算法来识别它。对于有大量证据的 $\mathcal{NP}$ 问题非确定性图灵机是有效设备。对于只有稀疏证据的 $\mathcal{NP}$ 问题，非确定性图灵机只是模型化了具有下列特性的一类判定性问题：给定一个证据，一个判定性问题的答案可以在多项式时间内进行验证。

$\mathcal{NP}$ 问题的一个证据由非确定性图灵机的计算树中的一个识别序列来刻画。

那么，不用证据，$\mathcal{NP}$中任意给定问题的确切复杂度是多少呢？这个问题的答案尚不清楚。不使用证据来求解$\mathcal{NP}$中任意问题的所有已知算法表明，它没有多项式界的时间复杂度。但到目前为止，还没有人证明这是必要的，即证明$\mathcal{P}\neq\mathcal{NP}$。同样，也没有人证明反面的情形，即证明$\mathcal{P}=\mathcal{NP}$。问题"$\mathcal{P}=\mathcal{NP}$?"是计算机科学中一个非常著名的公开问题。

如果求解问题P的任意算法A的复杂度开销都不小于B，则称B为问题P的复杂度下界。如果求解P的任意算法A的复杂度开销都不超过U，则称U为问题P的复杂度上界。

对$\mathcal{P}$中的任何问题，通常容易确定其复杂度下界，也就是说，给出精确的多项式界来表示求解该问题必需的步骤数量。图灵机 Div3 就是一个例子：它识别一个n比特输入串恰好需要n步。

对于$\mathcal{NP}$中的问题，总是难以确定其复杂度下界，甚至找一个新的较小的上界也很难。$\mathcal{NP}$问题的所有已知的复杂度界都是上界。例如，已经说明了$\lfloor\sqrt{N}\rfloor$是通过试除法来回答 SQUARE-FREENESS 问题的一个复杂度上界。本质上，上界说明求解这个问题只要这么多步骤就足够了，但在更少的步数内求解也是可能的。事实上，对于 SQUARE-FREENESS 问题，通过数域筛法来分解N的复杂度需要的步数小于$\lfloor\sqrt{N}\rfloor$，但仍然是一个上界。

对于以复杂度为安全性基础的现代密码学，确定$\mathcal{NP}$问题的非多项式下界的困难性有重要的影响。确定$\mathcal{NP}$问题的下界意味着计算复杂性理论的一个重大突破。

6.2.6 计算复杂性理论与现代密码学

基于复杂性理论的现代密码学将$\mathcal{P}\neq\mathcal{NP}$作为必要条件，称之为$\mathcal{P}\neq\mathcal{NP}$猜想。

另一方面，加密算法应向拥有正确的密钥的用户提供有效的加密/解密算法，而对于试图从密文中提取明文或者不用正确的密钥构造合法密文的人（攻击者或者密码分析者）来说，这是一个难处理的问题。因此，密钥对基于$\mathcal{NP}$问题的密码体制来说起着一个证据或者辅助输入的作用。

该猜想还构成了单向函数存在的必要条件。单向函数具有有趣而神奇的性质：对所有的输入x，计算$f(x)$是容易的，但是，除了可忽略的部分实例，给定绝大多数的$f(x)$值，要找出x是极为困难的。我们知道$\mathcal{NP}$难问题提供了实现具有这一性质的单向函数的候选问题。

进一步，单向函数的存在构成了数字签名存在的必要条件。数字签名应该具有这样的性质：验证一个签名的有效性是容易的，但是伪造一个合法的签名是困难的。

需要特别提及的是，$\mathcal{P}\neq\mathcal{NP}$猜想在密码学中有另一个令人着迷的应用：它在零知识证明协议和交互式证明系统中起着根本性的重要作用。

零知识证明协议是一种交互式程序，它运行在两个主体之间，分别称为证明者和验证者。验证者具有多项式界的计算能力，证明者拥有辅助输入，协议允许证明者向验证者证明自己知道一个$\mathcal{NP}$问题的 YES 答案，而且不让后者知道怎样进行这样的证明（即不向后者泄漏辅助输入）。这样，验证者对于证明者的辅助输入获得的是"零

知识”。这样一个证明可以用非确定性图灵机增加一条随机纸带来模拟。证明者可以利用辅助输入，可以指示图灵机关于输入问题沿着识别序列移动(证明 YES 答案)。证明的复杂度是关于输入规模的一个多项式。验证者需要向证明者提问，要求证明者指示图灵机或者沿着识别序列或者沿着另外一个不同序列移动，而提问的问题是均匀随机的。因此，在验证者看来，该证明系统正好按照随机化图灵机的模式运行，其差错概率通过独立重复执行可以减小到一个可忽略量，也正基于这一特性，验证者相信证明者确实知道输入问题 YES 答案。

$\mathcal{P}\neq\mathcal{NP}$猜想在零知识证明协议中起到以下两个作用：$\mathcal{NP}$问题的辅助输入使得证明者可以进行有效的证明；问题的困难性意味着验证者自己不能检验证明者的声明。

另一方面，即使密码体制基于$\mathcal{NP}$完全问题，$\mathcal{P}\neq\mathcal{NP}$猜想并没有提供密码体制安全性的充分条件。著名的背包问题的破解就是一个反例。

那么，为什么基于$\mathcal{NP}$问题的密码体制会被攻破呢？现在给出两个简要而清晰的解释。

首先，计算复杂性理论方法使用了全称量词“任意实例 $I\in L$”来限定复杂性类中的语言 L，这就导致了最坏情形的复杂性分析，即使一个问题只有可忽略的少数困难实例，该问题也被认为是困难的。相反，密码分析只要能够破坏不可忽略的比例的实例，就认为是成功的。这就是破解一个基于$\mathcal{NP}$完全问题的密码体制未必能求解其基础的$\mathcal{NP}$完全问题。显然，对度量密码体制安全性来说，最坏情形的复杂度准则是做不到的，也没有用处。

另一个原因是确定$\mathcal{NP}$问题新的上界的内在困难性。对于基于$\mathcal{NP}$难问题的密码体制的安全性基础，即使已经证明了其困难性，最好的情形也不过是基于一个公开问题，因为只知道该问题的一个复杂性上界。实际上，对于基于$\mathcal{NP}$问题的密码体制，甚至连其基础问题的困难性也没有明确地界定。

将复杂性理论中的困难性作为现代密码学的安全性基础是并不充分的。可能有许多实际的方式危及实际应用的密码系统，而这些危害可能和构成算法的安全性基础的数学难题没有什么关系。

6.3　归约方法和模型

在许多情况下，可以把问题 P_2 的一个实例 I 转换为问题 P_1 的一个实例 $f(I)$，通过求解 $f(I)$就得到了实例 I 的解。

定义 6.3.1(**多项式归约**)　设语言 $A,B\subseteq\{0,1\}^*$。如果存在一个确定性多项式时间可计算的函数 $f:\{0,1\}^*\to\{0,1\}^*$，使得对于任意的 $x\in\{0,1\}^*$，$x\in A$ 当且仅当 $f(x)\in B$，那么就称语言 A 可以多项式归约到语言 B，记为 $A\leqslant_m B$。f 称为从 A 到 B 的归约。

下面的简单引理是很重要的。表明：如果 $A\leqslant_m B$ 而且 B 是容易的，那么 A 也是容易的。

引理 6.3.1 如果 $A \leqslant_m B$，并且 $B \in P$，那么 $A \in P$。

证明：如果 $A \leqslant_m B$，并且 $B \in P$，则存在两个确定性图灵机 M 和 N，满足以下条件：

(1) M 计算某个函数 $f: \Sigma^* \to \Sigma^*$，使得 $x \in A \Leftrightarrow f(x) \in B$；

(2) 存在多项式 $p(n)$，使得 $T_M(n) \leqslant p(n)$；

(3) N 判定了语言 B；

(4) 存在多项式 $q(n)$，使得 $T_N(n) \leqslant q(n)$。

下面构造一个多项式时间的确定性图灵机来判定语言 A。给定输入 x，把 x 作为 M 的输入，得到 $f(x)$。然后把 $f(x)$ 输入 N，并根据 N 接受(拒绝)$f(x)$来决定接受(拒绝)x。

由于 M 计算了从语言 A 到 B 的一个归约，而 N 判定了语言 B，构造的新的确定性图灵机当然可以判定语言 A。下面说明其运行时间是多项式的。注意到，M 用来计算 $f(x)$的时间至多为 $p(n)$。进一步，$T_M(n) \leqslant p(n)$意味着$|f(x)| \leqslant p(n)+n$，这是由于机器 M 开始时纸带上的字符串长度为 n，而至多经过 $p(n)$步后停止，所以在停机后非空单元数不会超过 $p(n)+n$。因此，对于输入 $f(x)$，机器 N 运行的时间至多为 $q(p(n)+n)$。从而，新构造的机器的整个运行时间至多为 $p(n)+q(p(n)+n)$，仍然是输入规模 n 的多项式。这样，就证明了 $A \in P$。

显然，如果把 P 替换为 NP，类似的结论也是成立的。

引理 6.3.2 如果 $A \leqslant_m B$，并且 $B \in NP$，那么 $A \in NP$。

下面的结论表明，如果 B 至少与 A 一样困难，而 C 至少与 B 一样困难，那么C 至少与 A 一样困难。也就是说，关系$\leqslant_m$ 是可传递的。

引理 6.3.3 如果 $A \leqslant_m B$，$B \leqslant_m C$；那么 $A \leqslant_m C$。

6.3.1 非确定性多项式时间完备

尽管不清楚是否 $\mathcal{P}=\mathcal{NP}$，但是知道，$\mathcal{NP}$中某些问题和 $\mathcal{NP}$中任何问题的难度是一样的。在这个意义上，如果有一个有效算法求解其中的一个问题，那么 $\mathcal{NP}$中的任何问题都能够找到有效的求解算法。这种问题称为非确定性多项式时间完备的(NP 完全，简单记为 NPC)。

SATISFIABILITY(可满足性)问题是一个著名的 NP 完全问题，也是发现的第一个 NP 完全问题。设 $E(x_1,x_2,\cdots,x_n)$表示由 n 个布尔变量 $x_1,x_2,\cdots,x_n$ 使用布尔运算符，如 $\wedge$、$\vee$ 和 $\neg$，构成的一个布尔表达式。

问题 SATISFIABILITY。

输入 $X=(x_1,\neg x_1,x_2,\neg x_2,\cdots,x_n,\neg x_n)$

$E(x_1,x_2,\cdots,x_n)$

$E(x_1,x_2,\cdots,x_n)$的真值赋值是 X 的一个子列表 X'，满足：对于 $1 \leqslant i \leqslant n$，$X'$要么包含 x_i，要么包含 $\neg x_i$，但不能二者都包含，而且 $E(X')=\text{True}$。

问题 $E(x_1,x_2,\cdots,x_n)$可以满足吗？

也就是说，存在一个真值赋值吗？

如果给定一个可满足的真值赋值，那么显然 YES 回答可以在多项式时间内验证。因此，根据定义 6.2.3，SATISFIABILITY 属于 NP。注意，有 2^n 个可能的真值赋值，到目前为止，还没有确定性多项式时间的算法来确定它是否是可满足的。

关于 SATISFIABILITY 是 NP 完全的，是由 Cook 证明，该证明是构造性的，它将任意一个非确定性多项式时间图灵机归约为一个求解 SATISFIABILITY 的图灵机。

对于某一个 NP 完全问题，如果可以找到新的更小的上界，就可以多项式归约为整个 NP 类的新结果。因此，人们希望设计出安全性基于 NP 完全问题的密码算法，对于这类密码体制的成功攻击将会导致一类困难问题的解决，这是不可能的。然而，无论是实现一个安全使用的密码算法，还是通过对这种密码体制的攻击来求解整类 NP 完全问题，这样的想法到目前为止还没有富有成效的结果。

6.3.2 归约方法与可证明安全性理论

可证明安全性理论，就是在计算复杂性理论的框架下，利用“归约”的方法对密码体制进行严格的安全性证明，把一些已经得到深入研究的计算困难问题或者基本的密码模块归约到对安全协议的攻击。假如存在一个攻击者能够对安全协议或密码方案发动有效的攻击，那么就可以利用该攻击者构造一个算法用于求解困难问题。如果归约是多项式时间的，那么从计算复杂性理论的角度来说，对安全协议的攻击难度等同于求解该协议基于的困难问题或者密码模块，所以对于安全性来说，计算困难假设与安全的密码模块已经足够了。

在可证明安全性理论中，要证明一个密码方案 $\mathcal{C}$ 的安全性，一般做法是这样的：

(1) 根据密码方案的设计需求，形式地定义 $\mathcal{C}$ 的安全性。

(2) 选择一个难以求解的困难问题，或者难以破解的安全密码模块，这通常是所设计的密码方案的最基本的组成构件 $\mathcal{P}$。

(3) 把求解 $\mathcal{P}$ 的问题归约到攻破 $\mathcal{C}$，也就是说，给定 $\mathcal{P}$ 的一个实例 $\mathcal{I}_{\mathrm{P}}$，构造 $\mathcal{C}$ 的一个实例 $\mathcal{I}_{\mathrm{C}}$，并把 $\mathcal{C}(\mathcal{I}_{\mathrm{C}})$ 的解答转换成 $\mathcal{P}(\mathcal{I}_{\mathrm{P}})$ 的解答。这里要求问题 $\mathcal{C}$ 和 $\mathcal{P}$ 的实例构造，以及它们的解答的转换都是多项式时间的。

由于基本模块 $\mathcal{P}$ 是难以求解的计算困难问题或者安全的密码模块，通常经过了长期的深入研究，所以 $\mathcal{P}$ 的安全性保证了 $\mathcal{C}$ 是难以攻破的。

这类证明方法中，首先强调在一般框架中这样一个事实：允许敌手完全访问密码模块，但是以黑箱的方式。敌手可以对其选择进行任何查询，而黑箱总是能够正确地在常数时间内给予回答。

必须说明的是，可证明安全性理论也有局限性：首先要注意到模型的设计，即所建模型都涵盖了哪些攻击。这个模型并不考虑计时攻击，其中敌手试图从计算时间中提取秘密信息。一些其他的攻击通过分析计算所需的电磁能量来得到秘密，或者使得模块对于某些计算失败，本模型也没有涵盖这些攻击。但这并不意味着可证明安全的方案就一定不能抵抗这类攻击，而是说未证明可以抵抗这类攻击；其次即使应用具有可证明安全性的方案，也可能有其他方式破坏安全性：有时证明了安全性，但

问题可能是错误的,也可能应用了错误的模型或者协议被错误操作,甚至软件本身可能有“Bugs”。

另一需要注意的问题是基础假设的选取:可证明安全性是以某一假设为基础的,因此一旦该假设靠不住,安全性证明也就没有意义了(当然不一定意味着可构造对方案的攻击实例);选取基础假设的原则就是“越弱越好”,通常称弱假设为标准假设。基础假设的强弱是比较不同安全方案的重要尺度之一。

6.4 应用举例

设计与分析密码方案和安全协议是信息安全中的核心研究课题之一。传统的设计方法,是针对密码算法或安全协议的目标,基于某个困难问题假设或者基本密码模块提出密码算法或安全协议,并接受公众的安全性分析;如果经过长时间(如十几年)的分析仍然不能破译,就认为它是安全的,或者在发现某些安全漏洞之后,对其进行修补和改进。由于新的密码算法或安全协议分析技术的出现是不确定的,而任何新的分析技术都可能使得密码方案或安全协议宣告破解,所以传统的设计方法很难确保一个密码算法或安全协议的安全性。事实上,在多年来的信息安全实践中,一些一开始公认为很困难的密码体制被完全攻破(如基于背包问题的 Chor-Rivest 密码系统),一些公认为安全的协议被发现存在严重的安全漏洞(如基于 RSA-PKCS1.5 标准的 SSL 协议),因此,一时间内缺乏攻击绝不能就简单认定一个密码方案是安全可靠的。另一方面,尽管密码算法和安全协议是在一些困难问题或者 NP 完全问题的基础上提出来的,但其安全性不一定就等价于那些困难问题,要评估它的确切复杂性或者安全性到底有多高,需要进行严格的理论分析。

如何设计高度可信的安全协议和密码算法,是近年来的一个焦点研究问题。国内外很多学者从各方面作了尝试,其中可证明安全性理论就是为解决上述问题而提出的一种有效的解决方案。可证明安全性理论是一种公理化的设计思想,它可以保证:假如存在一个攻击者能够对安全协议或密码算法发动有效的攻击,那么就可以利用该攻击者构造一个算法用于求解困难问题或者破解密码模块。

可证明安全性理论的应用价值是显而易见的:可以把主要精力集中在基本密码模块和计算困难问题的研究上,这些是基础性的、带有艺术色彩的研究工作;作为安全协议基本组件或者密码方案的计算困难假设,如果这些基本模块是不安全的,所设计的方案一定是不安全的。另一方面,可证明安全性理论本质上是一种公理化研究方法,其最基础的假设是“好”的密码模块存在。

在可证明安全性理论研究的最初时期,研究工作集中于设法给出正确的安全性定义,然后设计能够满足该安全定义的安全协议或者密码方案。20 世纪 80 年代初,Goldwasser、Micali 和 Rivest 等首先比较系统地阐述了这一思想,给出了具有可证明安全性的加密和签名方案[8,9]。

早期的可证明安全性研究工作不依赖于其他的理想假设,称为标准模型下的可证明安全性理论。一方面,这些研究工作在理论上具有重要的意义,但是由此设计出

来的安全协议或密码算法却非常不实用，大多数情况下其可证明安全性是以损失效率为巨大代价的；另一方面，归约技巧来自于复杂性理论，其有效性是以多项式时间来衡量的，因此，标准模型下所得到的归约大多数是多项式时间的，这些结果只是在理论上达到了目的，但对于实际应用中的密码方案的安全性并无实质性的影响。所以，寻求更加高效的归约证明从而保证确切的安全性或者说实际的安全性，一直是众多学者的努力目标。

到20世纪90年代中期，人们开始研究"面向实际的可证明安全性"[1]，以提高安全协议的效率。主要思想就是在一些理想模型的理论框架下来处理问题，在这些模型中把某些具体的对象看作理想的随机函数，从而得到可证明安全而又高效实用的方案。第一个模型是由Fait和Shamir[7]引入，而由Bellare和Rogaway[3]正式提出的随机预言模型(Random Oracle Model，ROM)，把Hash函数作为理想的随机函数；也有人提出了理想密码模型(Ideal Cipher Model，ICM)，把分组密码作为真正的随机置换来处理可证明安全性的问题。

理想模型的引入，使得情况大为改观：一方面，过去仅作为纯粹理论研究的可证明安全性理论，迅速在实际应用领域取得重大进展，一大批快捷有效的安全方案相继提出；另一方面，人们开始在理想模型下考虑可证明安全性的"具体安全性"或者"确切安全性"：不再仅仅满足于多项式时间的安全性归约，而是可以给出较准确的安全性度量。

随机预言模型下的面向实际的可证明安全性理论既从理论上保证了密码方案的高度安全性，又能很容易地用于工程实践，目前已为国际密码学界和产业界广为接受，是可证明安全性理论最成功的实际应用。随机预言模型目前业已得到广泛认可，在该模型下通过可证安全理论研究设计密码算法与安全协议已成为工程实践中的一个准则，并且被近年来的标准化运动所采纳，包括RSA的PKCS标准、IEEE的P1363标准、NESSIE工程等。

6.4.1 归约效率与实际安全性

在20世纪80年代初期，一些具体的可证明安全的加密方案和签名方案相继提出。然而，这些证明只是具有理论上的影响，因为所提出的方案和给出的归约完全是不切实际的，当然，仍然是多项式的。从复杂性的角度来看，其归约的确是有效的(即多项式时间的)，因此对密码系统的多项式时间的攻击将导致一个多项式时间的算法用于破解基础计算假设。但是，后者尽管是多项式时间的，它可能需要几百年来破解一个实例。

举例来说，考虑基于大整数分解的密码协议。假设提供了一个从整数分解到攻击的归约，但是该归约导致的分解算法的复杂度是$2^{25}k^{10}$，其中k是要分解的整数的比特长度。这一结果的确与不存在多项式时间的分解算法的假设相矛盾。然而，对于一个1024比特的数($k=2^{10}$)，它所提供的算法需要2^{125}次基本操作，比当前最好的算法要复杂得多，如NFS算法只需要2^{100}次基本操作。所以，这一归约只对超过4096比特的数才有意义，因为此时$k=2^{12}$，$2^{145}<2^{149}$，其中2^{149}是使用最好的算法分

解 4096 比特的数的代价。

此外,人们提出的很多方案也不是实用的。事实上,这些协议是时间和存储空间的多项式,但并非足够高效以用于实际实现。

近些年来,人们尝试提出一些实用的方案,而且可以提供实际的归约和确切的复杂度,从而在合适的假设下对现实的参数提供安全性证明,即标准模型下(在复杂性框架下)的确切归约。例如,假设 1024 比特的整数不能在小于 2^{70} 个基本操作的情况下进行分解,密码协议不能在小于 2^{60} 个基本操作的情况下破解。

遗憾的是,在标准模型下,实际的归约很难与实际的方案共存。因此,需要对敌手做一些假设:攻击是一般性的,它与某些对象的具体实现是无关的。

(1) 杂凑函数,在“随机预言模型”下。

(2) 分组密码,在“理想密码模型”下。

(3) 代数群,在“一般群模型”下。

随机预言模型在密码领域已经被广泛接受。顺便说一下,人们发现了“一般群模型”下某些方案的缺陷,而“随机预言模型”也并不等价于标准模型。人们已经对此给出了几个差距[4]。但是,随机预言模型下的所有的反例都是不自然的、违反直觉的。因此,对于实际和自然的构造方案,主要在这一模型下分析其安全性。随机预言模型下的安全性证明至少对方案的安全性给出了很强的论断。此外,相对于使用强的计算假设在标准模型下给出的证明,使用弱的计算假设在随机预言模型下给出的证明更加具有实际意义。

6.4.2 随机预言模型

如前面所述,在 20 世纪 90 年代以前,标准模型下的可证明安全性理论是研究人员关注的焦点。然而,这方面的研究工作只是为可证明安全性研究奠定了理论基础,由此所设计出来的可证明安全性方案是以损失效率为代价的,直至今天,实用的标准模型下可证明安全方案也是少之又少。效率与可证明安全性似乎很难同时满足,效率最高的方案中没有一个在标准模型中得到了证明。

许多密码方案使用杂凑函数(如 MD5,或者 SHA-1、SHA-256、SHA-384、SHA-512)。使用杂凑函数的最初动机是希望用一个短的签名来签署长消息。为了达到抗抵赖的目的,对杂凑函数的最小需求是抗碰撞性,即签名者不可能找到两个不同的消息使得其杂凑值是一样的。后来人们意识到,杂凑函数不仅对于签名方案的安全性是一个本质的要素,而且对于大多数其他方案也是一样的。为了得到安全论断,同时保持使用杂凑函数而设计方案的效率,几个作者建议了这样的假设:杂凑函数的行为类似于一个随机函数。首先,Fiat 和 Shamir[7] 启发式地提出了一个与因子分解“一样安全”的签名方案。接着,Bellare 和 Rogaway 在 1993 年提出了随机预言模型(ROM)[3],宣告了设计实用可证明安全方案时代的到来。

在这个称为“随机预言模型”的方法中,杂凑函数被形式化为一个 Oracle,它对每一个新的查询产生一个真正随机的值。当然,如果用相同的查询询问两次,所得到的回答是一样的。这正好对应于复杂性理论中的“oracle”,因此其名字就是随机预言。

Bellare 和 Rogaway 在文献[3]中提出以下观点：假定各参与方(包括敌手)共同拥有一个公开的随机的预言(oracle)$\mathcal{R}$。当设计一个实际的协议 $\mathcal{P}$时，首先在随机预言模型下设计一个协议 $\mathcal{P}^{\mathcal{R}}$ 并证明其正确性，然后在实际方案中用“适当选择”的伪随机函数(如 Hash 函数)h 取代该预言。一般来说，这样设计出来的协议可以和当前协议的实现效率相当，比标准模型下的方案要有效得多，同时也保留了可证明安全性的优势。

在假定参与方拥有一个随机 oracle 的模型下证明一个协议的正确性，然后使用适当的密码模块(伪随机函数 PRF)来代替这个 oracle。然而，在用 PRF 来实现协议时，其种子是不能为敌手所知的，从而敌手不能访问 oracle。由于安全杂凑函数具有抗碰撞性，为了确保安全性和协议的实现效率，一些人主张采用 Hash 函数类似随机函数的假设来设计协议，因此它成为许多安全方案的重要组成部分。首次明确采用公开的随机预言模型，包括敌手在内的各参与方都可以访问该 oracle，是 Fait 和 Shamir 在文献[7]中提出的，他们使用这个模型把身份识别方案转换成签名方案。

假设提出一个协议问题 Π。为了设计一个安全协议 $\mathcal{P}$来解决该问题(要求该问题与模块 h 是无关的)，可按照以下步骤执行：

(1) 在随机预言计算模型中为 Π 建立形式的定义，各参与方(包括敌手)共享一个或多个随机 ordcle $\mathcal{R}$;

(2) 在随机预言模型中设计一个解决问题 Π 的有效协议 $\mathcal{P}$;

(3) 在随机预言模型中证明 $\mathcal{P}$满足 Π 的形式定义；

(4) 在实际应用中用 Hash 函数 h 代替 $\mathcal{R}$。

正确地执行这一方案，可以设计出安全而且有效的协议。实际上，到目前为止，按照这一方案构造的协议在实践中证明是安全的。但是，这里的可证明安全性断言都是在随机预言模型下的断言，而用杂凑函数来代替 oracle 只是来自于实践经验的一种探索。

应该注意到，这里随机 oracle(即 Hash 函数)对每一个新的询问产生一个随机数作为回答，但是如果对相同的值询问两次，那么得到的回答是相同的，这是和随机函数的微小区别。但这并未改变该方法论的成功，因为只要求在敌手看来像随机函数。

在随机预言模型方法论中，一般要求 Hash 函数 h 在设计上要能抵抗各种密码分析攻击，不会暴露某些相关数学结构。文献[3]中指出，选择 h 并不需要太麻烦，一个适当选择的杂凑函数就可以满足要求。尽管 MD5 及 SHA 本身不是一个好的选择，但只需截短其输出或用某种非标准方式使用，如 $h(x)=\text{MD5}(xx)$。

关于这一模型，没有人能够对其在实际中的有效性提出可信的反驳，而只是给出了一些理论上的反例，这些反例或者明显有设计错误的，或者是针对认为的安全概念的。因此，这个模型在该领域内得到了普遍接受，并被看作一个好的模型，用于分析实际的安全级别。即使它没有提供一个严格的安全性证明(如同标准模型那样，不依赖于任何理想假设)，它标明了：只要杂凑函数没有缺陷，该模型下的证明保证了方案设计的整体安全性。

这个模型也可以看作是对敌手能力的一种限制。实际上，它意味着其攻击是一

般性的，而不考虑杂凑函数的特定的实例化。因此，实际攻击需要使用杂凑函数的弱点或者特定的性质。把杂凑函数用另一个代替则可以避免攻击。

总之，正是由于该模型能够为非常有效的协议提供安全性验证，几乎所有的标准化组织都要求设计可证明安全的、至少是在随机预言模型下的方案。

6.4.3 计算假设

在密码学中，许多安全概念并不能在无条件的情况下得到保证。因此，安全性一般依赖于计算假设：单向函数的存在性、或者单向置换的存在性、或者是陷门单向函数（置换）的存在性。单向函数是一个满足以下条件的函数 f：任何人容易计算函数值，但是给定 $y=f(x)$，恢复 x（或者 y 的任何前像）在计算上是不可行的。单向置换是一个双射的单向函数。对于加密来说，希望只有接收者才可以求逆，于是陷门单向置换是一个特殊的单向置换，其秘密信息（即陷门）有助于对函数进行求逆。

给定计算假设“在没有陷门信息的情况下，计算函数的逆是不可行的”，希望不需要额外的假设即可得到安全性。形式上证明这一事实的唯一方法是证明“攻击密码协议的敌手可以用于构造算法，该算法能够求解基础计算假设”。

在计算假设之间存在一个偏序关系：如果问题 P 比问题 P' 更困难（P' 归约到 P），那么问题 P 的困难性假设就比问题 P' 的困难性假设弱。密码方案所需要的假设越弱，其安全性就越高。

在基于数论的公钥密码学中有两类主要的计算假设：

（1）基于整数分解的方案和基于 RSA 问题的方案。

（2）基于离散对数问题的方案和基于 Diffie-Hellman 问题的方案。第一个使用的群是 $\mathbb{Z}_p^*$ 的循环子群。现有许多方案是基于椭圆曲线的循环子群构造的。

1. 整数分解与 RSA 问题

最著名的困难问题是整数分解问题：把两个素数 p 和 q 相乘得到 $n=p\cdot q$ 是容易的，而把合数 n 分解为素因子 p 和 q 则不是一件简单的事情。目前，最有效的分解算法是数域筛法。数域筛法（NFS）是超多项式的、亚指数算法，其复杂度是

$$\mathcal{O}(\exp((1.923+o(1))(\ln n)^{1/3}(\ln\ln n)^{2/3}))$$

该方法在 1999 年 8 月创造了纪录，把一个 155 位的整数（512 比特）分解为两个 78 位素数的乘积，所分解的数称为 RSA-155，来自于“RSA 挑战列表”，用于衡量 RSA 密码体制的安全性，该密码体制在 SSL 握手协议的软件和硬件实现中广泛使用。

这一记录非常重要，因为 155 位对应于 512 比特，而这正是（在 Internet 的 SSL 实际部署中）RSA 密码体制实现中常用的规模。

$$\begin{aligned}\text{RSA}-155 &= 1094173864157052742180970732204035761200\backslash\\ &\quad 0373294544920599091384213147634998428893\backslash\\ &\quad 4784717997257891267332497625752899781833\backslash\\ &\quad 7970765372440271467435315933543338 97\\ &= 1026395928297411057720541965739916759007\backslash\\ &\quad 16567808038066803341933521790711307779\end{aligned}$$

* 106603488380168454820927220360012878679\
207958575989291522270608237193062808643

整数乘法只是提供了一个单向函数，没有任何可能来对其求逆。现在不知道如何使得整数分解更容易一些。有一些代数结构是基于整数 n 的分解，其中某些计算在不知道 n 的分解的情况下是困难的，在知道 n 的分解的情况下很容易，如有限商环 $\mathbb{Z}_n$，如果 $n=p\cdot q$，则同构于 $\mathbb{Z}_p\times\mathbb{Z}_q$。

例如，对任何元素 x，计算其 e 次幂是容易的。但是，要计算 e 根，看起来需要知道满足 $ed\equiv 1 \bmod \varphi(n)$ 的整数 d。这里 $\varphi(n)$ 是 Euler 函数，对于 $n=p\cdot q$ 的特殊情形，$\varphi(n)=(p-1)(q-1)$。因此，$ed-1$ 是 $\varphi(n)$ 的倍数，故等价于 n 的分解。

1978 年，Rivest、Shamir 和 Adleman 定义了著名的 RSA 问题并设计了首个公钥密码体制 RSA。

RSA 问题：设 $n=p\cdot q$ 是两个相同规模的大素数的乘积，e 是与 $\varphi(n)$ 互素的整数。

对给定的 $y\in\mathbb{Z}_n^*$，计算 y 的模 e 次根 x，即满足 $x^e=y \bmod n$ 的 $x\in\mathbb{Z}_n^*$。

设 $n=\prod p_i^{v_i}$，则 Euler 函数容易计算

$$\varphi(n)=n\times\prod\left(1-\frac{1}{p_i}\right)$$

因此，利用 n 的分解（陷门），RSA 问题很容易求解。但是没有人知道是否必须利用 n 的分解来求解 RSA 问题，更不知道如何在不知道 n 的分解的情况下来求解 RSA 问题。

RSA 假设：对任何两个足够大的素数的乘积 $n=p\cdot q$，RSA 问题是难解的（可能与分解 n 一样困难）。

2. 离散对数与 *Diffie-Hellman* 问题

设 $(\mathbb{G},\cdot)$ 表示一个阶为 q 的循环群，其中 q 是素数。令 g 表示 $\mathbb{G}$ 的一个生成元，即 $\mathbb{G}=\langle g\rangle$。与离散对数相关的困难问题，以简洁的形式描述如下：

离散对数问题（DL）：给定群 $y\in\mathbb{G}$，计算 $x\in\mathbb{Z}_q^*$，使得 $y=g^x$，记为 $x=\log_g y$。

计算 Diffie-Hellman 问题（CDH）：对于任意的整数 $a,b\in\mathbb{Z}_q^*$，给定 $\langle g,g^a,g^b\rangle$，计算 g^{ab}。

判定性 Diffie-Hellman 问题（DDH）：对于任意的整数 $a,b,c\in\mathbb{Z}_q^*$，给定 $\langle g,g^a,g^b,g^c\rangle$，判定是否有 $c\equiv ab \bmod q$ 成立。

上述问题显然是以从强到弱的顺序进行排列的，即 DL$\geqslant$CDH$\geqslant$DDH，$A\geqslant B$ 表示问题 A 至少与问题 B 一样困难。然而，在实际中，没有人知道如何求解其中的任何一个问题，除非可以破解 DL 问题本身。

而且，这些问题都是随机自归约的，即任何实例都可以归约到一个均匀分布的实例。例如，对于一个给定的元素 y，想要计算它对于基底 g 的离散对数 x。选择一个随机的 $t\in\mathbb{Z}_q$，计算 $z=ty$。那么 z 就是群中均匀分布的元素，而根据离散对数 $\alpha=\log_g z$ 就可以计算出 $x=\alpha/t$。因此，它们只有平均复杂情形，如果能够在多项式

时间内求解不可忽略的部分实例，那么就可以在期望多项式时间内求解任何实例。

目前，求解离散对数问题的最有效的算法依赖于其基础群。对于一般性的群（没有特定的代数性质可以应用），算法复杂度是 q 的平方根。但是，对于$\mathbb{Z}_p^*$ 的子群，存在一个更好的方法，即基于数域上的筛法，正如因子分解问题一样。一般数域筛法的复杂度是超多项式但亚指数的：

$$\mathcal{O}(\exp((1.923+o(1))(\ln p)^{1/3}(\ln\ln p)^{2/3}))$$

在 2001 年 4 月，该算法创造了最近的记录，对 120 位的素数 p 计算出了$\mathbb{Z}_p^*$ 中的离散对数。因此，只要一般攻击不能应用，而且其生成元的阶至少为 160 比特，512 比特的素数仍然是足够安全的。

对于签名方案只要求群的离散对数问题困难即可，而加密方案需要陷门，因而要求群中的某些 DH 问题也是难以求解的。

6.4.4 数字签名方案和公钥加密方案的概念与安全性定义

本节主要形式化地描述数字签名方案和公钥加密方案的定义，并给出这些方案想要达到的确切安全目标。

数字签名是用于电子文档的签名，对应于手写签名：用户对消息 m 的签名是一个字符串，它依赖于消息 m，依赖于用户特定的公开密钥和私有密钥，还可能依赖于随机数，任何人仅使用用户的公钥即可验证签名的有效性。直观上，数字签名方案（简称签名方案）的安全性有以下要求：如果没有用户的私钥，则不可能伪造用户的签名。给出签名方案的确切定义，以及对签名方案的可能攻击。

定义 6.4.1　一个签名方案 $S=(\mathcal{K},\mathcal{S},\mathcal{V})$ 由以下 3 个算法组成。

（1）密钥生成算法 $\mathcal{K}$：输入系统参数，输出一对匹配的公钥和私钥(pk,sk)。

（2）签名算法 $\mathcal{S}$：输入消息 m 和私钥 sk，输出签名 σ。

（3）验证算法 $\mathcal{V}$：输入签名 σ、消息 m 和公钥 pk，输出 1（表示签名有效）或 0（表示签名无效）。

下面刻画一些实际情形中可能出现的安全概念。一方面，敌手的目标有各种：

- 泄漏签名者的私钥，这是最严重的攻击，称为完全破解；
- 构造有效的算法，能够以高的概率来签署消息，也称为通用伪造；
- 提出一个新的消息——签名对，称为存在性伪造。对应的安全级别称为存在性非伪造。

在许多情形下，存在性伪造攻击并不是危险的，因为其输出的消息可能是没有意义的。然而，存在性可伪造的签名方案本身并不能保证签名者的身份。例如，它不能用于证实看起来随机的元素，如密钥。进一步，它不能形式地保证抗抵赖的性质，因为任何人都可能产生一个消息并提供有效的签名。

另一方面，敌手可以使用不同的手段来进行伪造。人们主要关注针对签名方案的两种特定攻击类型：无消息攻击和已知消息攻击。在无消息攻击中，敌手只知道签名者的公钥。在已知消息攻击中，敌手可以访问有效的消息——签名对列表。根据该列表创建的方式，可区分不同的情况，其中最强的定义是适应性选择消息攻击，

这里攻击者可以要求签名者对其选择的任何消息进行签名，以适应性的方式：敌手可以根据前面的回答而改变其查询的内容。

如果签名方案不是确定性的，一个消息可对应于多个签名。于是上述一些定义就变得有歧义了。首先，对于已知消息攻击，存在性伪造就变成伪造一个在攻击过程中没有得到的新的消息/签名对的能力，这一点比较微妙。实际上采取较强的规则，要求敌手伪造一个消息的签名，而该消息的签名没有查询过。在宽松的规则下，攻击者只要对给定消息输出第二个签名，该签名不同于以前得到的对同一个消息的签名，则攻击者就成功，这一规则称为延展性，而相应的安全级别称为非延展性。同样地，在选择消息攻击下，敌手可以对同一个消息进行几次查询，每一个新的回答给予他某些信息。稍微弱一点儿的安全模型，称为单次选择消息攻击，允许敌手对每个消息至多进行一次签名查询，即敌手不能把同一个消息提交两次进行签名查询。

在设计签名方案时，希望在适应性选择消息攻击下可以抵抗存在性伪造攻击，甚至达到非延展性。确切的定义[3]如下。

设$(\mathcal{K},\mathcal{S},\mathcal{V})$是一个签名方案；敌手 $\mathcal{A}$ 拥有公钥，可以适应性地选择 w 个消息并进行签名查询，设查询签名的消息为 $m_1,\cdots,m_w$。如果 $\mathcal{A}$ 能够伪造一个新消息 $m(m\notin\{m_1,\cdots,m_w\})$的签名的概率是可忽略的，即

$$\mathrm{Succ}_{\mathcal{S}}^{\mathrm{euf}}(\mathcal{A})=\Pr\left[(pk,sk)\leftarrow\mathcal{K}(1^k),(m,\sigma)\leftarrow\mathcal{A}^{\mathcal{S}_{sk}}(pk):\mathcal{V}(pk,m,\sigma)=1\right]$$

是可忽略的，则称该签名方案在适应性选择消息攻击下是存在性非伪造的。

注意到，由于敌手可以发起适应性选择消息攻击，签名算法是可以得到的，没有任何限制，因此标记为 $\mathcal{A}^{\mathcal{S}_{sk}}$。当然，敌手的回答有一个自然的限制，即输出的消息——签名对不曾从签名 oracle $\mathcal{S}_{sk}$ 中得到（非延展性），或者输出的消息也没有查询过（存在性非伪造）。

公钥加密方案的目的是允许任何知道 Alice 公钥的人都可以向她发送消息，使得 Alice 是唯一可以进行恢复的人。

定义 6.4.2　一个公钥加密方案 $S=(\mathcal{K},\mathcal{E},\mathcal{D})$由以下 3 个算法组成。

（1）密钥生成算法 $\mathcal{K}$：给定输入 1^k（其中 k 是安全参数），算法 $\mathcal{K}$ 产生一对匹配的公钥和私钥(pk,sk)。算法 $\mathcal{K}$ 是概率性的。

（2）加密算法 $\mathcal{E}$：给定消息 m 和公钥 pk，$\mathcal{E}$ 产生 m 的密文 c。该算法可以是概率性的，记为 $\mathcal{E}_{pk}(m;r)$，其中 r 是算法 $\mathcal{E}$ 的随机输入。

（3）解密算法 $\mathcal{D}$：给定密文 c 和私钥 sk，$\mathcal{D}_{sk}(c)$返回明文 m。该算法是确定性的。

对于公钥加密方案来说，敌手的目标可以有多种。公钥加密方案的首个安全概念是单向性：仅使用公开的数据，敌手不能得到给定密文的对应明文。更确切地，对任何敌手来说，不使用私钥而对函数 $\mathcal{E}$ 求逆的成功概率在概率空间 $\mathcal{M}\times\Omega$ 上以及敌手的内部随机掷币是可忽略的，其中 $\mathcal{M}$ 是消息空间，Ω 是加密方案使用的随机数 r 的空间：

$$\mathrm{Succ}_{\mathcal{S}}^{\mathrm{ow}}(\mathcal{A})=\Pr_{m,r}\left[(pk,sk)\leftarrow\mathcal{K}(1^k):\mathcal{A}(pk,\mathcal{E}_{pk}(m;\sigma))=m\right]$$

然而，许多应用需要对加密方案有更高的要求，即语义安全性（IND）[8]，也称为不可区分性：攻击者获知明文的一些信息，如针对关键查询的回答“yes”或者“no”。在这

一安全定义中，敌手选择两个消息，对其中之一进行加密，敌手以明显大于一半的概率区分出该密文是对哪一个消息加密，在计算上是不可行的，形式化的定义如下：

$$\mathrm{adv}_{\mathcal{A}} = 2 \times \Pr_{b,r}\left[\begin{array}{l}(pk, sk) \leftarrow \mathcal{K}(1^k), (m_0, m_1, s) \leftarrow \mathcal{A}_1(pk), \\ c = \mathcal{E}_{pk}(m_b; r): \mathcal{A}_2(m_0, m_1, s, c) = b\end{array}\right] - 1$$

是可忽略的，其中敌手可看作一个两阶段的攻击者 $\mathcal{A}=(\mathcal{A}_1, \mathcal{A}_2)$。

另一个定义是非延展性：给定密文，敌手试图产生一个新的密文，使得其明文是语义相关的。这一概念比上面的语义安全性更强，但是在大多数有趣的情况下(如选择密文攻击)二者都是等价的。因此，只关注单向性和语义安全性。

另一方面，攻击者可以进行多种形式的攻击，根据其所得到的信息：由于考虑的是非对称加密，攻击者可以自己选择并加密任何明文，即选择明文攻击。它也可以得到其他的信息，由下面的部分或者全部 oracle 来表示：

(1) 有效性检验 oracle：输入密文 c，回答其是否是一个有效的密文。如此一个弱的预言，已经被用于攻击一些著名的密码方案。

(2) 明文检验 oracle：输入一对(m, c)，其输出回答了 c 是否加密了 m。这一攻击称为明文检验攻击。

(3) 解密 oracle：对任何密文，输出相应的明文。当然，自然要求不能对该预言询问挑战密文。

如果一旦挑战密文给定，对这些 oracle 的访问就受限了，此类攻击称为非适应性选择密文攻击，因为查询不能依赖于挑战密文，但是可以依赖于之前的查询。相反，如果在挑战密文给定后，对这些 oracle 的访问不受限，则此类攻击称为适应性的攻击。这一区别在选择密文攻击中广泛使用：非适应性选择密文攻击(CCA1)，也称为午餐攻击，以及适应性选择密文攻击(CCA2)。后者当然是最强的攻击，将在本章中称为选择密文攻击。

6.4.5 RSA-FDH 签名方案

1996 年，Bellare 和 Rogaway 提出了经典的全域杂凑(Full-Domain Hash)签名方法论，并结合 RSA 假设，提出了著名的 RSA-FDH 签名方案，该方案由以下 3 个算法组成。

(1) 密钥生成算法：输入 1^k，该算法随机选择两个 $k/2$ 比特的素数 p 和 q，计算 $n=p \cdot q$；随机选择 $e \in \mathbb{Z}_{\varphi(n)}^*$ 并计算 d 使得 $ed \equiv 1 \bmod \varphi(n)$。

用户的公钥为(e, n)，私钥为(d, n)。

设 $H: \{0,1\}^* \to \mathbb{Z}_n^*$ 是一个抗碰撞的密码杂凑函数。

(2) 签名算法：给定消息 m 和私钥(d, n)，该算法计算并输出签名

$$\sigma = H(m)^d \bmod n$$

(3) 验证算法：给定签名 σ、消息 m 和公钥(e, n)，该算法检验

$$\sigma^e = H(m) \bmod n$$

如果成立，输出 1(签名有效)，否则输出 0 (签名无效)。

为了证明 RSA-FDH 的安全性，先给出 RSA 问题的定量描述：

对于所有的 $k\in\mathbb{N}$，如果一个求逆算法 $\mathcal{I}$ 可以在 $t(k)$ 时间内以 $\varepsilon(k)$ 的成功概率破解 RSA 问题，则称算法 $\mathcal{I}$ 可以 (t,ε) 破解 RSA 问题。

如果任何求逆算法都不能 (t,ε) 破解 RSA 问题，则称 RSA 是 (t,ε) 安全的。

定理 6.4.1　假设 RSA 问题是 (t',ε') 安全的，则 RSA-FDH 签名方案是 (t,ε) 安全的，满足

$$t = t' - (q_{\mathrm{h}} + q_{\mathrm{sig}} + 1)\cdot \mathcal{O}(k^3)$$

$$\varepsilon = \frac{1}{\left(1-\dfrac{1}{q_{\mathrm{sig}}+1}\right)^{q_{\mathrm{sig}}+1}}\cdot q_{\mathrm{sig}}\cdot \varepsilon',$$

其中，q_{sig} 表示签名询问的次数，q_{h} 表示随机预言询问的次数。当 q_{sig} 较大时，有

$$\varepsilon \approx \exp(1)\cdot q_{\mathrm{sig}}\cdot \varepsilon'$$

证明：假设伪造者 $\mathcal{F}$ 经过 q_{sig} 次签名询问和 q_{h} 次随机预言询问，可以 (t,ε) 攻破 RSA-FDH 签名方案。构造一个求逆算法 $\mathcal{I}$ 可以 (t',ε') 破解 RSA 困难问题。

假设给定了实例 y，挑战者要找到满足 $x^e=y \bmod n$ 的 x。注意到，在随机预言模型方法论中，伪造者 $\mathcal{F}$ 不能自己计算 Hash 值，只能通过 oracle 询问来得到消息的杂凑值。

算法 $\mathcal{I}$ 设置用户的公钥为 (n,e)，以此来运行算法 $\mathcal{F}$。攻击者 $\mathcal{F}$ 需要访问签名 oracle 和随机 oracle，$\mathcal{I}$ 需要自己回答这些 oracle。为了简单起见，当 $\mathcal{F}$ 询问一个消息的签名时，假定其已经对该消息进行了相应的 Hash 询问。如果没有，$\mathcal{I}$ 自己进行 Hash 询问。$\mathcal{I}$ 使用一个计数器 i，初始化为 0。

算法 $\mathcal{I}$ 按照以下方式来回答伪造者 $\mathcal{F}$ 对消息 m_i 的 oracle 询问：

当 $\mathcal{F}$ 对消息 m 进行 Hash oracle 询问时，$\mathcal{I}$ 把计数器 i 增加 1，令 $m_i=m$，并随机选择 $r_i\in\mathbb{Z}_n^*$。$\mathcal{I}$ 依概率 p 返回 $h_i=r_i^e \bmod n$，依概率 $1-p$ 返回 $h_i=yr_i^e \bmod n$，这里 p 是一个固定的概率，其值将在以后确定。

当 $\mathcal{F}$ 对消息 m 进行签名询问时，它已经对 m 进行了 Hash 询问，于是 m 等于某个 m_i。如果 $h_i=r_i^e \bmod n$，则 $\mathcal{I}$ 返回 r_i 作为签名。由于 $h(m_i)=h_i=r_i^e \bmod n$，显然 r_i 就是 m_i 的有效签名。否则，算法中止，求逆算法失败。

最终，算法 $\mathcal{F}$ 中止并输出一个伪造 (m,σ)。假设 $\mathcal{F}$ 之前已经对 m 进行了 Hash 询问。如果没有，$\mathcal{I}$ 自己进行 Hash 询问。无论何种情况，存在某个 i 使得 $m=m_i$。那么，如果 $h_i=yr_i^e \bmod n$，则有 $\sigma=h_i^d=y^d r_i \bmod n$，于是 $\mathcal{I}$ 输出

$$x = y^d = \sigma/r_i$$

作为 y 的逆。由于 $y=x^e \bmod n$，x 即为所求。否则，算法中止，$\mathcal{I}$ 失败。

算法 $\mathcal{I}$ 能够回答所有的签名询问的概率至少为 $p^{q_{\mathrm{sig}}}$，然后 $\mathcal{I}$ 输出 y 的逆的概率为 $1-p$。所以，$\mathcal{I}$ 至少以概率 $\alpha(p)=p^{q_{\mathrm{sig}}}\cdot(1-p)$ 输出 y 的逆。容易看出，当 p 取 $p_{\max}=1-1/(q_{\mathrm{sig}}+1)$ 时，$\alpha(p)$ 取最大值，并且

$$\alpha(p_{\max}) = \frac{1}{q_{\mathrm{sig}}}\left(1-\frac{1}{q_{\mathrm{sig}}+1}\right)^{q_{\mathrm{sig}}+1}$$

从而，有

$$\varepsilon(k)=\frac{1}{\left(1-\frac{1}{q_{\mathrm{sig}}+1}\right)^{q_{\mathrm{sig}}+1}}\cdot q_{\mathrm{sig}}\cdot\varepsilon'(k)$$

而当 q_{sig} 较大时，有 $\varepsilon\approx\exp(1)\cdot q_{\mathrm{sig}}\cdot\varepsilon'$。

算法 $\mathcal{I}$ 的运行时间等于算法 $\mathcal{F}$ 的运行时间加上计算 h_i 所需要的时间，于是得到时间 t 的公式。

注意到 FDH 类型的签名方案是确定性的签名方案，每一个消息都有唯一的签名，所以在其安全性证明中，一个模拟器要么能够产生消息的签名，要么就不能产生其签名，这在一定程度上限制了安全性归约。这一发现推动了概率签名方法的设计思想，如 PSS 签名方法，其中每一个消息都有多个签名。

在可证明安全的签名方案研究中，Pointcheval 等人提出来的 Folklore 引理是对完善随机预言模型方法论下签名方案的归约证明的一个重要贡献，适用于许多基于身份识别协议的签名方案。Folklore 引理的基本思想是 oracle 重放攻击，即在随机预言模型方法论的归约证明中，重放多项式个不同的随机 oracle(相当于为敌手提供多个模拟环境)，如果敌手能以不可忽略的概率伪造签名，就可以求解该方案的基础困难问题，如离散对数问题。该方法的缺点是所得到的归约不够紧。

6.4.6 Cramer-Shoup 公钥加密方案

Cramer 和 Shoup[5] 于 1998 年提出了第一个比较实际的标准模型下可证明安全的公钥加密方案，该方案的困难假设是判定性 Diffie-Hellman 问题。由于其安全性归约是在标准的杂凑函数假设(抗碰撞)下得到的，并不依赖于随机预言模型，所以受到了很大的关注。

设 G 是有限域 Z_p^* 的阶为 q 的子群，p 和 q 为素数，且 $q\mid p-1$，g_1 和 g_2 是 G 中两个随机的非单位元的元素。设 $x=(x_1,x_2)$，$y=(y_1,y_2)$，$z=(z_1,z_2)$ 表示在 $1\sim q-1$ 之间的数对；$g=(g_1,g_2)$，$u=(u_1,u_2)$ 表示 G 中的元素对；r 是 $1\sim q-1$ 之间的随机数，记 $g^x=g_1^{x_1}g_2^{x_2}$，$g^{rx}=g_1^{rx_1}g_2^{rx_2}$。假设 H 是合适的抗碰撞杂凑函数。

(1) 密钥生成算法：随机选取 $g_1,g_2\in G$，$x_1,x_2,y_1,y_2,z_1,z_2\in Z_q$，计算

$$c=g^x,\quad d=g^y,\quad h=g^z$$

用户 Alice 的私钥为 $(x_1,x_2,y_1,y_2,z_1,z_2)$，公钥为 (g_1,g_2,c,d,h,H)。

(2) 加密算法：为了发送消息 $m\in G$，选择一个随机数 r，令

$$u_1=g_1^r,\quad u_2=g_2^r,\quad e=h^rm,$$

然后计算

$$\alpha=H(u_1,u_2,e),\quad v=c^rd^{r\alpha}$$

密文就是四元组 (u_1,u_2,e,v)。

(3) 解密算法：要解密 (u_1,u_2,e,v)，Alice 首先计算 $\alpha=H(u_1,u_2,e)$，然后利用她的私钥计算 $u^{x+\alpha y}$ 并验证这个结果是否等于 v(因为 $u^{x+\alpha y}=g^{rx+r\alpha y}=c^rd^{r\alpha}$)。如果它不等于 v，拒绝该消息；如果通过这个检验，则继续进行解密：把 e 除以 u^z，因为 $u^z=g^{rz}=h^r$，而 $e=h^rm$，所以这就是明文 m。

对于 Cramer-Shoup 加密方案，如果存在一个适应性选择密文攻击的敌手 $\mathcal{A}$ 能够破坏其语义安全性，那么就可以构造一个算法 $\mathcal{B}$ 来求解判定性 Dffie-Hellman 问题：即判断一个四元组 (g_1, g_2, u_1, u_2) 是否满足 Diffie-Hellman 性质 $\log_{g_1} u_1 = \log_{g_2} u_2$。

定理 6.4.2　如果判定性 Diffie-Hellman 问题是难解的，那么 Cramer-Shoup 公钥加密方案在适应性选择密文攻击下是语义安全的。

证明：假设算法 $\mathcal{B}$ 给定了一个四元组 (g_1, g_2, u_1, u_2)。首先 $\mathcal{B}$ 随机选取 x, y, z，令

$$c = g^x, \quad d = g^y, \quad h = g^z$$

并把 (c, d, h) 以及 $g = (g_1, g_2)$ 作为公钥发送给 $\mathcal{A}$。

$\mathcal{B}$ 需要回答 $\mathcal{A}$ 的解密询问，为了回答对 (u_1', u_2', e', v') 的询问，$\mathcal{B}$ 计算 $\alpha' = H(u_1', u_2', e')$，如果 $v' \neq u'^{x+\alpha' y}$，则拒绝该消息，否则解密 w'/u'^z 返回给 $\mathcal{A}$。

当敌手 $\mathcal{A}$ 输出两个明文 m_0 和 m_1 用于区分测试时，$\mathcal{B}$ 随机选择 $b \in \{0, 1\}$，设置密文为 $y^* = (u_1, u_2, e, v)$，其中

$$e = u^z m_b, \quad v = u^{x+\alpha y}, \quad 而\ \alpha = H(u_1, u_2, e)$$

敌手 $\mathcal{A}$ 需要判断 y^* 是 m_0 还是 m_0 的加密。

如果 (g_1, g_2, u_1, u_2) 满足 Diffie-Hellman 性质，那么 y^* 就是 m_b 的真实加密，其中 $r = \log_{g_1} u_1 = \log_{g_2} u_2$ 就是加密所用的随机数，所以敌手 $\mathcal{A}$ 将以明显大于 1/2 的概率正确地猜测到 b 值。

如果 (g_1, g_2, u_1, u_2) 不满足 Diffie-Hellman 性质，那么 $\mathcal{A}$ 所看到的信息与 b 是无关的，所以他只能有 1/2 的机会猜测到 b 值。所以，通过使用不同的 x, y, z 值多次运行这个模拟攻击，如果 $\mathcal{A}$ 大多数情况下都能正确地猜测到 b 值，$\mathcal{B}$ 就确定 (g_1, g_2, u_1, u_2) 满足 Diffie-Hellman 性质。下面两个引理完成了归约证明。

引理 6.4.1　如果算法 $\mathcal{B}$ 的输入是 Diffie-Hellman 四元组，那么敌手的视图与比特 b 的联合分布与实际攻击中的分布是不可区分的。

证明：设 (g_1, g_2, u_1, u_2) 是一个 Diffie-Hellman 四元组，考虑敌手的视图与比特 b 的联合分布。设 $u_1 = g_1^r, u_2 = g_2^r$。

在这种情况下，显然加密 oracle 的输出分布是正确的。为了完成证明，需要论证解密 oracle 的分布也是正确的。如果 $\log_{g_1} u_1' = \log_{g_2} u_2'$，则称 (u_1', u_2', e', v') 是一个有效的密文。

注意到，如果密文是有效的，而 $u_1' = g_1^{r'}, u_2' = g_2^{r'}$，那么 $h^{r'} = (u_1')^{z_1}(u_2')^{z_2}$，因此，解密 oracle 输出 $e'/h^{r'}$，这正是所期望的。根据下面的断言，引理 6.4.1 即得证。

断言：对于密码体制的实际攻击和对于模拟情况下的攻击，除了可忽略的概率之外，解密 oracle 都拒绝所有无效的密文。

下面证明断言。根据敌手的视图，考虑四元组 $P = (x_1, x_2, y_1, y_2) \in \mathbb{Z}_q^4$ 的分布。设 $w = \log_{g_1} g_2$。

根据敌手的视图，P 是由下面的两个超平面相交的平面上一个随机点：

$$\log_{g_1} c = x_1 + w x_2 \qquad (6.1)$$

$$\log_{g_1} d = y_1 + wy_2 \tag{6.2}$$

这两个方程容易由公钥推出。加密 oracle 的输出并没有对 P 进一步限制，因为

$$\log_{g_1} v = rx_1 + wrx_2 + \alpha ry_1 + \alpha rwy_2 \tag{6.3}$$

假设敌手提交了一个无效的密文(u_1', u_2', e', v')给解密 oracle，其中 $\log_{g_1} u_1' = r_1'$，$\log_{g_1} u_2' = wr_2'$并且 $r_1' \neq r_2'$。解密 oracle 将会拒绝该密文，除非 P 恰好位于式(6.4)定义的超平面 H 上：

$$\log_{g_1} v' = r'_1 x_1 + wr'_2 x_2 + \alpha' r'_1 y_1 + \alpha' r'_2 wy_2 \tag{6.4}$$

其中 $\alpha' = H(u_1', u_2', w')$。注意到方程(6.1)、方程(6.2)和方程(6.4)是线性无关的，H 与 P 相交于一条直线。

对于第一次提交的无效密文，解密 oracle 将以 $1-1/q$ 的概率拒绝该密文，对于第 i 次提交的无效密文，解密 oracle 将至少以 $1-1/(q-i+1)$ 的概率拒绝该密文。因此，除了可忽略的概率，解密 oracle 将拒绝所有的无效密文。

引理 6.4.2 如果算法 $\mathcal{B}$的输入是一个随机的四元组，那么比特 b 的分布与敌手的视图是无关的。

设 $u_1 = g_1^{r_1}, u_2 = g_2^{wr_2}$，而且不妨假设 $r_1 \neq r_2$，因为二者相等的概率是可忽略的。根据下面的两个断言，引理 6.4.2 得证。

断言 1：如果解密 oracle 拒绝所有的无效密文，那么比特 b 的分布与敌手的视图是无关的。

为了证明这一断言，考虑点 $Q = (z_1, z_2)$。在攻击开始时，这是位于公钥确定的直线

$$\log_{g_1} h = z_1 + wz_2 \tag{6.5}$$

上的一个随机点。如果解密 oracle 只对有效的密文(u_1', u_2', e', v')进行解密，则敌手得到的只是线性无关的关系 $r'\log_{g_1} h = r'z_1 + r'wz_2$，不会泄漏 Q 的信息。

下面分析加密 oracle 输出的(u_1, u_2, e, v)。有 $e = \varepsilon m_b$，其中 $\varepsilon = u_1^{z_1} u_2^{z_2}$。考虑方程

$$\log_{g_1} \varepsilon = r_1 z_1 + wr_2 z_2 \tag{6.6}$$

显然，式(6.5)和式(6.6)是线性无关的，于是，相对于 b 以及除了 e 之外的敌手的视图，ε 是均匀分布的。换言之，ε 是一个完善的一次一密。因此 b 是与敌手的视图无关的。

断言 2：除了可忽略的概率，解密 oracle 拒绝所有的无效密文。

为了证明断言 2，与引理 6.4.1 一样，在敌手视图的条件下考虑 $P = (x_1, x_2, y_1, y_2)$的分布。根据敌手的视图，这是位于由超平面(1)、(2)和下面的超平面交叉的直线上的一个随机点：

$$\log_{g_1} v = r_1 x_1 + wr_2 x_2 + \alpha r_1 y_1 + \alpha r_2 wy_2 \tag{6.7}$$

方程(6.7)来自于解密 oracle 的输出。

假设敌手提交了一个无效密文$(u_1', u_2', e', v') \neq (u_1, u_2, e, v)$，其中

$$\log_{g_1} u_1' = r_1', \quad \log_{g_1} u_2' = r_2', \quad r_1' \neq r_2'$$

设 $\alpha' = H(u_1', u_2', e')$。

考虑下面 3 种情况。

情况 1：$(u_1', u_2', e') = (u_1, u_2, e)$。

在这种情况下，杂凑值是相同的，但是 $v' \neq v$，所以解密 oracle 当然会拒绝它。

情况 2：$(u_1', u_2', e') \neq (u_1, u_2, e)$，$\alpha' \neq \alpha$。

除非点 P 位于方程(6.4)定义的超平面 H，解密 oracle 将拒绝。但是，方程(6.1)、方程(6.2)、方程(6.7)和方程(6.4)是线性无关的，这可以通过下面式子进行验证

$$\det \begin{pmatrix} 1 & w & 0 & 0 \\ 0 & 0 & 1 & w \\ r_1 & wr_2 & \alpha r_1 & \alpha w r_2 \\ r_1' & wr_2' & \alpha' r_1' & \alpha' w r_2' \end{pmatrix} = w^2 (r_2 - r_1)(r_2' - r_1')(\alpha' - \alpha) \neq 0$$

因此，H 与直线 L 相交与一点，由此可知解密 oracle 拒绝，除了可忽略的概率之外。

情况 3：$(u_1', u_2', e') \neq (u_1, u_2, e)$，$\alpha' = \alpha$。

这一情况发生的概率是可忽略的，否则与杂凑函数的抗碰撞性质矛盾。

6.5　注记

本章重点介绍了信息安全研究中常用的计算复杂性理论的基本概念、基本原理、典型的归约方法和模型，同时介绍了计算复杂性方法与技术在密码算法和安全协议设计与分析中的应用。

计算复杂性理论是一门发展相对比较成熟的学科，有着丰富的研究成果和广泛的应用，感兴趣的读者可参阅文献[2]。与现代密码学密切相关的一些方法和技术在文献[10]中介绍得比较详细，很值得一读。关于计算复杂性方法和技术在信息安全领域中的应用文献很多，如文献[1]、文献[3～9]，还可以参阅文献[6]所引用的一些参考文献。

参考文献

[1] Mihir Bellare. Practice-Oriented Provable-Security. In: E. Okamoto, G. Davida and M. Mambo ed. Proceedings of First International Workshop on information Security (ISW97), LNCS 1396. Berlin: Springer Verlag, 1998, pp. 221-231

[2] Arora S, Barak B. Computational Complexity: A Modern Approach. Cambridge University Press, 2009

[3] Bellare M, Rogaway P. Random Oracles are practical: A Paradigm for Designing Efficient Protocols. In: Proceedings of the First ACM Conference on Computer and Communications Security, Fairfax, Virginia, USA, 1993, pp 62-73

[4] Canetti R, Goldreich O, Halevi S. The Random Oracle Methodology, Revisited. In: Proceedings of 30th Annual ACM Symposium on the Theory of Computing, ACM, 1998, pp 209-218

[5] Cramer R, Shoup V. A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack, Advances in Cryptology-Crypto 1998, volume 1462 of Lecture

Notes in Computer Science, pp 13-25, 1998

[6] Feng Dengguo. Research on Theory and Approach of Provable Security, Journal of Software, 16(10)1743-1756, 2005

[7] Fiat A, Shamir A. How to prove yourself: practical solutions to identification and signature problems. In: A. Odlyzko ed, Advances in Cryptology-crypto′86 Proceedings, LNCS 263, Berlin: Springer-Verlag, 1987, pp. 186-194

[8] Goldwasser S, Micali S. Probabilistic Encryption. Journal of Computer and System Science, 1984, 28:270-299

[9] Goldwasser S, Micali S, Rivest R. A Digital Signature Scheme Secure Against Adaptive Chosen-Messege Attacks. SIAM Journal of Computing, 1988,17(2): 281-308

[10] Mao W B. Modern Cryptography: Theory and Practice. Prentice Hall PTR, 2003

第 7 章　数理统计方法与技术

数理统计学是研究大量随机现象规律性的一门数学学科。它以概率论为基础，从实际观测资料出发，研究如何用有效的方法去收集和利用数据资料，对随机变量的分布函数、数字特征等进行估计、分析和推断。数理统计在信息安全领域有着广泛的应用，如密码学、数字水印及网络入侵检测等。本章的重点是介绍一些在信息安全研究中常用的数理统计方法和技术，内容包括数理统计的基本概念、典型的估计方法、假设检验及典型的应用实例。

7.1　基本概念

本节主要介绍数理统计中需要用到的一些基本概念，包括总体与样本、统计量与抽样分布，另外还有一些常用统计量的分布。

7.1.1　总体与样本

在数理统计中，称研究对象的全体为总体。组成总体的每一个元素称为个体。例如，如果想考察某学校学生的体重情况，则该学校学生体重的全体称为一个总体，而每一个学生的体重是一个个体。由总体的部分个体构成的集合称为总体的一个样本。在现实生活中，一般都是通过对样本进行观察来推断总体的性质。总体和样本是数理统计中的两个基本概念。

从数学角度来讲，任何一个总体，都可以用一个随机变量 X 来描述，它的取值在客观上有一定的分布。对总体的研究就是对随机变量 X 的概率分布的研究。X 的分布函数和数字特征分别称为总体的分布函数和数字特征。

定义 7.1.1　样本是由 n 个随机变量 X_i 组成的 n 维随机变量 $(X_1,X_2,\cdots,X_n)$，其中 X_i 表示由总体 X 取出的个体。每次抽取的数据是 n 维随机变量的一个值，可用 $(x_1,x_2,\cdots,x_n)$ 表示，它称为样本 $(X_1,X_2,\cdots,X_n)$ 的一个观测值，简称样本观测值。

定义 7.1.2　对于样本 $(X_1,X_2,\cdots,X_n)$，若 $X_1,X_2,\cdots,X_n$ 相互独立且每个 $X_i(i=1,2,\cdots,n)$ 与总体 X 有相同的概率分布，则称 $(X_1,X_2,\cdots,X_n)$ 为来自总体 X 的容量为 n 的简单随机样本。如无特别说明，数理统计所研究的样本一般都是指简单随机样本。

7.1.2　统计量与抽样分布

通过抽样观察，人们可以推断总体的一些性质。但在实际应用中，往往并不是利用样本本身的观测数据进行推断，而是针对不同的问题构造样本的某种函数，然后利用这些函数对总体的性质进行推断。数理统计称这种函数为统计量。

定义 7.1.3 设$(X_1, X_2, \cdots, X_n)$为总体 X 的一个样本,若 $g(X_1, X_2, \cdots, X_n)$是$(X_1, X_2, \cdots, X_n)$的一个函数,并且 g 中不含有任何未知参数,则称 $g(X_1, X_2, \cdots, X_n)$是一个统计量。

由于样本$(X_1, X_2, \cdots, X_n)$是 n 维随机变量,所以统计量 $g(X_1, X_2, \cdots, X_n)$也是随机变量。

定义 7.1.4 设$(x_1, x_2, \cdots, x_n)$是样本$(X_1, X_2, \cdots, X_n)$的观察值,则称 $g(x_1, x_2, \cdots, x_n)$是 $g(X_1, X_2, \cdots, X_n)$的观察值。

统计量是随机变量,它具有确定的概率分布,其分布称为抽样分布。若总体的分布函数是已知的,则抽样分布是确定的。一般情况下,抽样分布是很难精确计算出来的。只有在极少数情况下,如总体分布为正态分布下,某些抽样分布可能会有精确结果。下面将介绍一些常用统计量的分布。

7.1.3 常用统计量分布

1. χ^2 分布

定理 7.1.1 设 $X_1, X_2, \cdots, X_n$ 是来自总体 $N(0,1)$的样本,则称随机变量

$$\chi^2 = X_1^2 + X_2^2 + \cdots + X_n^2$$

服从自由度为 n 的χ^2分布,记为$\chi^2 \sim \chi^2(n)$。

2. t 分布

定理 7.1.2 设 $X \sim N(0,1)$,$Y \sim \chi^2(n)$,并且 X、Y 独立,则称随机变量

$$t = \frac{X}{\sqrt{Y/n}}$$

服从自由度为 n 的 t 分布。记为 $t \sim t(n)$。

3. F 分布

定理 7.1.3 设 $X \sim \chi^2(m)$,$Y \sim \chi^2(n)$,并且 X、Y 独立,则称随机变量

$$F = \frac{X/m}{Y/n}$$

服从自由度为(m,n)的 F 分布,记为 $F \sim F(m,n)$。其中 m 称为第一自由度,n 称为第二自由度。显然,如果 $X \sim F(m,n)$,则$\frac{1}{X} \sim F(n,m)$。

7.2 典型的参数估计方法

参数估计是数理统计中非常重要的一类问题。在实际问题中,当总体分布部分未知或完全未知时,人们往往会通过抽取样本来估计总体未知参数的值或关于总体的某些数字特征,这被称为参数估计问题。参数估计分为点估计和区间估计两种。下面将主要介绍 3 种典型的点估计方法(矩估计、极大似然估计和贝叶斯估计)及区间估计。

7.2.1　矩估计法

当总体 X 为连续型随机变量，设其概率密度为 $f(x;\theta_1,\theta_2,\cdots,\theta_k)$；当总体 X 为离散型随机变量时，$P\{X=x\}=p(x;\theta_1,\theta_2,\cdots,\theta_k)$。其中 $\theta_1,\theta_2,\cdots,\theta_k$ 为未知参数。

现从总体 X 中抽取样本 $X_1,X_2,\cdots,X_n$，由辛钦大数定律知，对任意 $\varepsilon>0$，有

$$\lim_{n\to\infty}P\left(\left|\frac{1}{n}\sum_{i=1}^{n}X_i-E(X)\right|<\varepsilon\right)=1$$

该定理可以推广为，对任意 $\varepsilon>0$，有

$$\lim_{n\to\infty}P\left(\left|\frac{1}{n}\sum_{i=1}^{n}X_i^l-E(X^l)\right|<\varepsilon\right)=1,\quad l=1,2,\cdots,k$$

定义 7.2.1　当 n 较大时，$E(X^l)\approx\frac{1}{n}\sum_{i=1}^{n}X_i^l, l=1,2,\cdots,k$，其中 $E(X^l)$ 与 X 的分布有关，因此是参数 $\theta_1,\theta_2,\cdots,\theta_k$ 的函数，从而可以形成一个包含 k 个未知参数 $\theta_1,\theta_2,\cdots,\theta_k$ 的联立方程组。该方程组的解 $\hat{\theta}_1,\hat{\theta}_2,\cdots,\hat{\theta}_k$ 分别作为参数 $\theta_1,\cdots,\theta_k$ 的估计量，称为矩估计量。该方法称为矩估计法。

例 7.2.1　设总体 X 服从正态分布 $N(\mu,\sigma^2)$，现从总体 X 中抽取样本 X_1，$X_2,\cdots,X_n$。试求未知参数 u 和 σ^2 的矩估计。

由于 $E(X)=\mu, E(X^2)=D(X)+[E(X)]^2=\sigma^2+\mu^2$，因此根据矩估计法有

$$\begin{cases}\mu=\dfrac{1}{n}\sum\limits_{i=1}^{n}X_i\\ \sigma^2+\mu^2=\dfrac{1}{n}\sum\limits_{i=1}^{n}X_i^2\end{cases}$$

解以上方程组得

$$\begin{cases}\hat{\mu}=\dfrac{1}{n}\sum\limits_{i=1}^{n}X_i\\ \hat{\sigma}^2=\dfrac{1}{n}\sum\limits_{i=1}^{n}X_i^2-\left(\dfrac{1}{n}\sum\limits_{i=1}^{n}X_i\right)^2\end{cases}$$

7.2.2　极大似然估计法

设总体 X 的概率分布为 $P(x;\theta_1,\theta_2,\cdots,\theta_k)$，其中 $\theta_1,\theta_2,\cdots,\theta_k$ 为未知参数。

定义 7.2.2　现从总体 X 中抽取样本 $X_1,X_2,\cdots,X_n$，$x_1,x_2,\cdots,x_n$ 是相应于样本 $X_1,X_2,\cdots,X_n$ 的一个样本值，则事件 $\{X_1=x_1,X_2=x_2,\cdots,X_n=x_n\}$ 发生概率为 $\prod_{i=1}^{n}P(x_i;\theta_1,\theta_2,\cdots,\theta_k)$。在这里将这个联合分布记为 $L(\theta_1,\theta_2,\cdots,\theta_k)$，其概率随 $\theta_i(i=1,2,\cdots,k)$ 的取值而发生变化，称为样本的似然函数。

定义 7.2.3　极大似然估计法就是固定样本观察值 $x_1,x_2,\cdots,x_n$，在 $\theta_i(i=1,2,\cdots,k)$ 的取值的可能范围内挑选 $\hat{\theta}_1,\hat{\theta}_2,\cdots,\hat{\theta}_k$ 使得 $L(\hat{\theta}_1,\hat{\theta}_2,\cdots,\hat{\theta}_k)=\max\limits_{\theta}L(\theta_1,$

$\theta_2,\cdots,\theta_k$)。这样得到的 $\hat{\theta}_1,\hat{\theta}_2,\cdots,\hat{\theta}_k$ 与样本值 $x_1,x_2,\cdots,x_n$ 有关,记为 $\hat{\theta}(x_1,x_2,\cdots,x_n)$,它被称为参数 $\theta_1,\theta_2,\cdots,\theta_k$ 的极大似然估计值。相应的统计量 $\hat{\theta}(X_1,X_2,\cdots,X_n)$被称为参数 $\theta_1,\theta_2,\cdots,\theta_k$ 的极大似然估计量。

若想求解 $\hat{\theta}(x_1,x_2,\cdots,x_n)$,很多情况下可先对似然函数取对数 $\ln L(\theta_1,\theta_2,\cdots,\theta_k)$,然后对各参数 $\theta_1,\theta_2,\cdots,\theta_k$ 分别求偏导数,并令它们为 0,即

$$\frac{\partial \ln L(\theta_1,\theta_2,\cdots,\theta_k)}{\partial \theta_i}=0 \quad i=1,2,\cdots,k$$

通过求解方程组可以得出 $\hat{\theta}_1,\hat{\theta}_2,\cdots,\hat{\theta}_k$。将 $\hat{\theta}_1,\hat{\theta}_2,\cdots,\hat{\theta}_k$ 表达式中的 $x_1,x_2,\cdots,x_n$ 均换为 $X_1,X_2,\cdots,X_n$,即得极大似然估计。

例 7.2.2 设总体 X 服从正态分布 $N(\mu,\sigma^2)$,现从总体 X 中抽取样本 $X_1,X_2,\cdots,X_n$。试求未知参数 u 和 σ^2 的极大似然估计。

因为总体 X 服从正态分布 $N(\mu,\sigma^2)$,则它的概率密度函数为

$$p(x)=\frac{1}{\sqrt{2\pi}\sigma}\exp\left[-\frac{1}{2\sigma^2}(x-\mu)^2\right]$$

设 $x_1,x_2,\cdots,x_n$ 为样本 $X_1,X_2,\cdots,X_n$ 的一组观测值,则似然函数为

$$L(u,\sigma^2)=\prod_{i=1}^{n}\frac{1}{\sqrt{2\pi}\sigma}\exp\left[-\frac{1}{2\sigma^2}(x_i-\mu)^2\right]=(2\pi\sigma^2)^{-\frac{n}{2}}\exp\left[-\frac{\sum\limits_{i=1}^{n}(x_i-\mu)^2}{2\sigma^2}\right]$$

因此

$$\ln L(\mu,\sigma^2)=-\frac{n}{2}\ln(2\pi)-\frac{n}{2}\ln\sigma^2-\frac{1}{2\sigma^2}\sum_{i=1}^{n}(x_i-\mu)^2$$

对 $\ln L(\mu,\sigma^2)$分别求关于 μ、σ^2 的偏导数,并令其为 0

$$\begin{cases}\dfrac{\partial \ln L(\mu,\sigma^2)}{\partial \mu}=\dfrac{1}{\sigma^2}\sum\limits_{i=1}^{n}(x_i-\mu)=0\\[2ex]\dfrac{\partial \ln L(\mu,\sigma^2)}{\partial \sigma^2}=-\dfrac{n}{2\sigma^2}+\dfrac{1}{2\sigma^4}\sum\limits_{i=1}^{n}(x_i-\mu)^2=0\end{cases}$$

解这方程组得 μ、σ^2 的极大似然估计为

$$\hat{\mu}=\frac{1}{n}\sum_{i=1}^{n}X_i,\quad \hat{\sigma}^2=\frac{1}{n}\sum_{i=1}^{n}X_i^2-\left(\frac{1}{n}\sum_{i=1}^{n}X_i\right)^2$$

7.2.3 贝叶斯估计

前面所说的矩估计或极大似然估计,都有一个共同的特点:即在抽取样本之前,对未知参数 θ 没有任何了解,所有信息均来自样本。贝叶斯估计的基本思想是,在进行抽样之前,已对参数 θ 有一定的知识,称为先验知识。这种先验知识用 θ 的某种概率分布来表达,称为 θ 的"先验分布",令 $h(\theta)$表示 θ 的密度函数。

设总体的概率密度函数为 $f(x,\theta)$(若总体为离散型的,称为概率函数),现从总

体 X 中抽取样本 $X_1,X_2,\cdots,X_n$，则在给定 θ 值的情况下样本的密度函数为 $f(X_1,\theta)\cdots f(X_n,\theta)$。而$(\theta,X_1,X_2,\cdots,X_n)$的联合密度为

$$h(\theta)f(X_1,\theta)f(X_2,\theta)\cdots f(X_n,\theta)$$

$(X_1,X_2,\cdots,X_n)$的边缘密度为

$$p(X_1,X_2,\cdots,X_n)=\int h(\theta)f(X_1,\theta)f(X_2,\theta)\cdots f(X_n,\theta)\mathrm{d}\theta$$

因此，在给定样本 $X_1,X_2,\cdots,X_n$ 的条件下，θ 的条件密度为

$$h(\theta\mid X_1,X_2,\cdots,X_n)=h(\theta)f(X_1,\theta)f(X_2,\theta)\cdots f(X_n,\theta)/p(X_1,X_2,\cdots,X_n)$$

以上密度函数包含了 θ 的先验知识与样本信息，称为 θ 的"后验分布"。在后验分布的基础上，可以对参数进行估计。如对点估计来说，常用的办法是取后验分布的均值作为 θ 的估计。

例 7.2.3　设总体 X 服从正态分布 $N(\theta,1)$，其中 θ 的先验分布为 $N(\mu,\sigma^2)$(μ 和 σ^2 为已知)。现从总体 X 中抽取样本 $X_1,X_2,\cdots,X_n$，求 θ 的贝叶斯估计。

因为总体 X 服从正态分布 $N(\theta,1)$及 θ 的先验分布为 $N(\mu,\sigma^2)$，则有

$$h(\theta)=(\sqrt{2\pi}\sigma)^{-1}\exp\left[-\frac{1}{2\sigma^2}(\theta-\mu)^2\right]$$

$$f(x,\theta)=(\sqrt{2\pi})^{-1}\exp\left[-\frac{1}{2}(x-\theta)^2\right]$$

因此，θ 的后验密度为

$$h(\theta\mid X_1,X_2,\cdots,X_n)=(\sqrt{2\pi}\sigma)^{-1}\exp\left[-\frac{1}{2\sigma^2}(\theta-\mu)^2\right]\Big/ I$$

其中 I 是一个与 θ 无关而只与 μ、σ^2、X_1、X_2、$\cdots$、X_n 有关的数。

$$-\frac{1}{2\sigma^2}(\theta-\mu)^2-\frac{1}{2}\sum_{i=1}^{n}(X_i-\theta)^2=-\frac{1}{2\eta^2}(\theta-t)^2+J$$

其中，

$$t=(n\overline{X}+\mu/\sigma^2)/(n+1/\sigma^2)$$

$$\eta^2=1/(n+1/\sigma^2)$$

在这里，J 与 θ 无关，因此，

$$h(\theta\mid X_1,X_2,\cdots,X_n)=I_1\exp\left[-\frac{1}{2\eta^2}(\theta-t)^2\right]$$

其中，$I_1=I\mathrm{e}^J$ 与 θ 无关，因为

$$\int_{-\infty}^{\infty}h(\theta\mid X_1,X_2,\cdots,X_n)\mathrm{d}\theta=1$$

所以 $I_1=(\sqrt{2\pi}\eta)^{-1}$，$\theta$ 的后验分布为正态分布 $N(t,\eta^2)$，因此 θ 的贝叶斯估计为

$$\tilde{\theta}=t=\frac{n}{n+1/\sigma^2}\overline{X}+\frac{1/\sigma^2}{n+1/\sigma^2}\mu$$

7.2.4　区间估计

前面所介绍的 3 种估计方法都是关于参数的点估计方法，即用一个数去估计未知参数。参数的点估计是具有确定形式的估计量，可以根据样本的观察值计算出参

数的确定估计值。但由于点估计量是一个随机变量,因而取参数真值的概率较低。与此同时,点估计量没有涉及与真值之间误差的界限问题。这就使得点估计方法具有一定的局限性。为解决这些问题,本节将讨论参数的区间估计方法。直观地讲,区间估计就是用一个区间去估计未知参数,把参数值估计在两个界限之间。

定义 7.2.4 设总体 X 的分布函数为 $F(x,\theta)$,θ 是未知参数。$(X_1,X_2,\cdots,X_n)$ 为总体 X 的样本。若存在两个统计量 $\hat{\theta}_1(X_1,X_2,\cdots,X_n)$ 与 $\hat{\theta}_2(X_1,X_2,\cdots,X_n)$,使得对一切样本,均有 $\hat{\theta}_1<\hat{\theta}_2$,并且

$$P(\hat{\theta}_1<\theta<\hat{\theta}_2)=1-\alpha,\quad 0<\alpha<1$$

则称区间$(\hat{\theta}_1,\hat{\theta}_2)$为参数 θ 的**置信区间**,称概率 $1-\alpha$ 为置信区间$(\hat{\theta}_1,\hat{\theta}_2)$的**置信度**,而 $\hat{\theta}_1$、$\hat{\theta}_2$ 分别称为**置信下限和置信上限**。

$1-\alpha$ 表明置信区间的可靠性,$1-\alpha$ 越接近于 1 就表示区间的可靠性越高。在固定 α 的情况下,$\hat{\theta}_2-\hat{\theta}_1$ 反映置信区间的精度,$\hat{\theta}_2-\hat{\theta}_1$ 越小,置信区间的精度就越大。

下面将讨论 3 种正态总体参数的置信区间。设总体 X 服从正态分布 $N(\mu,\sigma_0^2)$,$X_1,X_2,\cdots,X_n$ 是来自总体 X 的样本。

1. 已知方差 $\sigma^2=\sigma_0^2$、均值参数 μ 的置信区间

令 $\overline{X}=(X_1+X_2+\cdots+X_n)/n$,构造

$$U=\frac{\overline{X}-\mu}{\sigma_0}\sqrt{n}$$

它服从 $N(0,1)$分布。对给定的置信度 $1-\alpha$,查正态分位表得 $\mu_{\frac{\alpha}{2}}$,即

$$P\left(-\mu_{\frac{\alpha}{2}}<\frac{\overline{X}-\mu}{\sigma_0}\sqrt{n}<\mu_{\frac{\alpha}{2}}\right)=1-\alpha$$

因此

$$P\left(\overline{X}-\mu_{\frac{\alpha}{2}}\frac{\sigma_0}{\sqrt{n}}<\mu<\overline{X}+\mu_{\frac{\alpha}{2}}\frac{\sigma_0}{\sqrt{n}}\right)=1-\alpha$$

所以,μ 的置信度为 $1-\alpha$ 的置信区间为

$$\left(\overline{X}-\mu_{\frac{\alpha}{2}}\frac{\sigma_0}{\sqrt{n}},\overline{X}+\mu_{\frac{\alpha}{2}}\frac{\sigma_0}{\sqrt{n}}\right)$$

例 7.2.4 设某种电子管寿命服从 $N(\mu,40^2)$分布。现从大批电子管中抽取 100 只,算得平均寿命为 1000h,求整批电子管平均寿命的 95%的置信区间。

由于已知方差,所以电子管平均寿命置信区间为

$$\left(\overline{X}-\mu_{\frac{\alpha}{2}}\frac{\sigma_0}{\sqrt{n}},\overline{X}+\mu_{\frac{\alpha}{2}}\frac{\sigma_0}{\sqrt{n}}\right)$$

在这里,$\bar{x}=1000$,$n=100$,$\alpha=0.05$,$\sigma_0=40$。查正态分布分位数表得 $\mu_{\frac{\alpha}{2}}=1.96$。所以整批电子管平均寿命的 95%的置信区间为

$$\left(1000-1.96\times\frac{40}{\sqrt{100}},1000+1.96\times\frac{40}{\sqrt{100}}\right)$$

即(992.16,1007.84)。

2. 未知方差、均值参数 μ 的置信区间

令 $\overline{X}=(X_1+X_2+\cdots+X_n)/n, S_n=\sqrt{\dfrac{1}{n-1}\sum_{i=1}^{n}(X_i-\overline{X})^2}$，构造

$$T=\frac{\overline{X}-\mu}{S_n}\sqrt{n}$$

它服从自由度为 $n-1$ 的 t 分布。对给定的置信度 $1-\alpha$，自由度 $n-1$，查 t 分布分位表得到 $t_{\frac{\alpha}{2}}$，即

$$P\left(-t_{\frac{\alpha}{2}}<\frac{\overline{X}-\mu}{S_n}\sqrt{n}<t_{\frac{\alpha}{2}}\right)=1-\alpha$$

因此

$$P\left(\overline{X}-t_{\frac{\alpha}{2}}\frac{S_n}{\sqrt{n}}<\mu<\overline{X}+t_{\frac{\alpha}{2}}\frac{S_n}{\sqrt{n}}\right)=1-\alpha$$

所以，μ 的置信度为 $1-\alpha$ 的置信区间为

$$\left(\overline{X}-t_{\frac{\alpha}{2}}\frac{S_n}{\sqrt{n}},\overline{X}+t_{\frac{\alpha}{2}}\frac{S_n}{\sqrt{n}}\right)$$

例 7.2.5 某车间生产轴套，其直径 X 服从正态分布。从某日产品中随机抽查 6 个，测得直径为(单位：mm)：

14.6　15.1　14.9　14.8　15.2　15.1

试求直径均值的 95%的置信区间。

由于方差未知，所以直径均值置信区间为

$$\left(\overline{X}-t_{\frac{\alpha}{2}}\frac{S_n}{\sqrt{n}},\overline{X}+t_{\frac{\alpha}{2}}\frac{S_n}{\sqrt{n}}\right)$$

又因为

$$\bar{x}=(14.6+15.1+14.9+14.8+15.2+15.1)/6=14.95$$

$$\begin{aligned}S_n^2&=[(14.6-14.95)^2+(15.1-14.95)^2+(14.9-14.95)^2\\&\quad+(14.8-14.95)^2+(15.2-14.95)^2+(15.1-14.95)^2]/5\\&=(0.35^2+0.15^2+0.05^2+0.15^2+0.25^2+0.15^2)/5\\&=0.051\end{aligned}$$

$$S_n=\sqrt{0.051}\approx 0.2258$$

$n=6,\alpha=0.05$，查 t 分布分位数表得 $t_{0.025}(5)=2.571$。

所以直径均值置信区间为

$$\left(14.95-2.571\times\frac{0.2258}{\sqrt{6}},14.95+2.571\times\frac{0.2258}{\sqrt{6}}\right)$$

即(14.713,15.187)。

3. 未知均值、方差参数 σ^2 的置信区间

令 $\overline{X}=(X_1+X_2+\cdots+X_n)/n$，构造

$$\chi^2 = \frac{\sum_{i=1}^{n}(X_i - \overline{X})^2}{\sigma^2}$$

它服从自由度为 $n-1$ 的 $\chi^2(n-1)$ 分布。对给定的置信度 $1-\alpha$，查自由度为 $n-1$ 的 $\chi^2(n-1)$ 分布上侧分位数表得 $\chi^2_{1-\frac{\alpha}{2}}$ 和 $\chi^2_{\frac{\alpha}{2}}$，即

$$P\left(\chi^2_{1-\frac{\alpha}{2}} < \frac{\sum_{i=1}^{n}(X_i - \overline{X})^2}{\sigma^2} < \chi^2_{\frac{\alpha}{2}}\right) = 1-\alpha$$

因此

$$P\left(\frac{\sum_{i=1}^{n}(X_i - \overline{X})^2}{\chi^2_{\frac{\alpha}{2}}} < \sigma^2 < \frac{\sum_{i=1}^{n}(X_i - \overline{X})^2}{\chi^2_{1-\frac{\alpha}{2}}}\right) = 1-\alpha$$

所以，σ^2 的置信度为 $1-\alpha$ 的置信区间为

$$\left(\frac{\sum_{i=1}^{n}(X_i - \overline{X})^2}{\chi^2_{\frac{\alpha}{2}}}, \frac{\sum_{i=1}^{n}(X_i - \overline{X})^2}{\chi^2_{1-\frac{\alpha}{2}}}\right)$$

例 7.2.6 在例 7.2.5 中，试求直径方差 σ^2 的 95%的置信区间。

因为均值未知，所以 σ^2 的置信区间为

$$\left(\frac{\sum_{i=1}^{n}(X_i - \overline{X})^2}{\chi^2_{\frac{\alpha}{2}}}, \frac{\sum_{i=1}^{n}(X_i - \overline{X})^2}{\chi^2_{1-\frac{\alpha}{2}}}\right)$$

查自由度为 5 的 χ^2 分布表得 $\chi^2_{1-\frac{\alpha}{2}}(5)=0.831$，$\chi^2_{\frac{\alpha}{2}}(5)=12.833$。因此直径方差 σ^2 的 95%的置信区间为

$$\left(\frac{5\times 0.051}{12.833}, \frac{5\times 0.051}{0.831}\right)$$

即(0.01987,0.30686)。

7.3 假设检验

假设检验问题是数理统计的另一类重要问题。为了推断总体的某些性质，假设检验首先提出一些关于总体的假设，然后根据样本对所提出的假设做出判断，即是接受假设还是拒绝假设。假设检验分为两大类：一类是参数检验，该类检验是只知总体分布的类型，但不知其参数的情况；另一类是非参数检验，该类检验是在总体分布未知的情况下，对总体的性质进行假设检验。

7.3.1 基本原理

假设检验的基本思想是，首先提出关于总体性质的一些假设，称为原假设。然后

在原假设的条件下导出结论，若结论发生的概率很大，则认为原假设成立，反之若概率非常小，则否定原假设。该思想源于实践中被广泛采用的一条原则，即小概率事件在一次观察中是不会出现的。关于小概率事件发生的概率称之为显著性水平，用 α 来表示，它表示了假设检验的严格程度。α 越小，则否定原假设的说服力越强。通常情况下，α 会取 0.05、0.01 或 0.1。

基于以上基本思想，对于参数假设检验来说，基本步骤可以归纳为以下几步。

(1) 根据问题的要求提出原假设 H_0（或零假设）及备择假设 H_1，在这里备择假设 H_1 通常是原假设 H_0 的反面。

(2) 根据原假设 H_0 确定合适的检验统计量及分布。

(3) 确定显著性水平 α。

(4) 确定拒绝域的形式，按照 $P\{$拒绝 $H_0 | H_0$ 为真$\}=\alpha$ 计算拒绝域。

(5) 根据样本观察值计算统计量的具体值，若落入拒绝域中，则在显著性水平 α 条件下拒绝原假设 H_0，否则就接受原假设 H_0。

由于正态总体在现实生活中分布广泛，下面将重点介绍一下关于正态总体的一些参数假设检验方法。

7.3.2　单个正态总体的假设检验

1. 已知方差的单个正态总体的均值检验

设 $X_1,X_2,\cdots,X_n$ 是从正态总体 $N(\mu,\sigma_0^2)$ 中抽取的样本，其中 σ_0^2 是已知常数，欲假设检验 $H_0:\mu=\mu_0$，$H_1:\mu\neq\mu_0$。

在这里可用 $\overline{X}=(X_1+X_2+\cdots+X_n)/n$ 来构造统计量进行假设检验，在 H_0 成立时，$U=\dfrac{\overline{X}-\mu_0}{\sigma_0}\sqrt{n}$ 服从标准正态分布，因此 U 可作为检验的统计量。在给定显著性水平 α 的情况下，查正态分布表计算 $\mu_{\frac{\alpha}{2}}$，使

$$P\{|U|>\mu_{\frac{\alpha}{2}}\}=\alpha$$

从而检验的拒绝域为

$$W=\{|U|>\mu_{\frac{\alpha}{2}}\}$$

由样本 $X_1,X_2,\cdots,X_n$ 的观测值计算 U 的观测值 u，若 $|\mu|>\mu_{\frac{\alpha}{2}}$，则拒绝原假设 H_0，否则接受原假设 H_0。

这种检验法被称为 μ 检验法。

例 7.3.1　已知某种零件的尺寸服从正态分布，方差 $\sigma^2=1.21$，对一批这样零件检查 6 件，尺寸数据分别为（单位：mm）：

32.56　29.66　31.64　30.00　31.87　31.05

当显著性水平 $\alpha=0.05$ 时，能否认为此批零件的平均尺寸为 32.50mm？

设总体 X 表示该种零件的尺寸，则 X 服从正态分布 $N(\mu,1.21)$。问题可以归纳为对假设

$$H_0:\mu=32.50\quad H_1:\mu\neq 32.50$$

作假设检验。由于方差已知,故可用 u 检验法。首先计算

$$\bar{x} = (32.56 + 29.66 + 31.64 + 30.00 + 31.87 + 31.05)/6 = 31.13$$

然后可以计算统计量 U 的观测值

$$U = \frac{\bar{x} - \mu_0}{\sigma_0}\sqrt{n} = \frac{31.13 - 32.5}{\sqrt{1.21}}\sqrt{6} \approx -3.05$$

对给定的显著性水平 $\alpha = 0.05$,查正态分布表得临界值 $\mu_{\frac{\alpha}{2}} = \mu_{0.025} = 1.96$。因为 $|\mu| = 3.05 > 1.96$,所以拒绝原假设 H_0。即认为此批零件的平均尺寸不是 32.50mm。

2. 方差未知时单个正态总体的均值检验

设 $X_1, X_2, \cdots, X_n$ 是从正态总体 $N(\mu, \sigma_0^2)$ 中抽取的样本,其中 σ^2 是未知参数,欲假设检验 $H_0: \mu = \mu_0$, $H_1: \mu \neq \mu_0$。

因为 σ^2 是未知参数,所以不能利用前面的 μ 检验法。现选择以下统计量

$$T = \frac{\overline{X} - \mu_0}{S_n}\sqrt{n}$$

其中 $S_n = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(X_i - \overline{X})^2}$。当 H_0 成立时,T 服从自由度为 $n-1$ 的 t 分布。在给定显著性水平 α 的情况下,查 t 分布表计算 $t_{\frac{\alpha}{2}}$,使

$$P\{|T| > t_{\frac{\alpha}{2}}\} = \alpha$$

从而检验的拒绝域为

$$W = \{|T| > t_{\frac{\alpha}{2}}\}$$

由样本 $X_1, X_2, \cdots, X_n$ 的观测值计算 T 的观测值 t,若 $|t| > t_{\frac{\alpha}{2}}$,则拒绝原假设 H_0,否则接受原假设 H_0。

这种检验法被称为 t 检验法。

例 7.3.2 有容量为 50 而方差未知的正态总体的样本。若 $\bar{x} = 2.7$,且 $\sum_{i=1}^{50}(x_i - \bar{x})^2 = 113$,当显著性水平 $\alpha = 0.05$ 时,试检验假设 $H_0: \mu = 3$。

因为正态总体 X 的方差未知,所以要使用 t 检验法,由所得数据可算出

$$s_n = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2} = \sqrt{\frac{113}{50-1}} \approx 1.519$$

$$t = \frac{\bar{x} - \mu_0}{s_n}\sqrt{n} = \frac{2.7 - 3}{1.519}\sqrt{50} \approx -1.397$$

对给定的显著性水平 $\alpha = 0.05$,查 t 分布的分位数表得临界值 $t_{\frac{\alpha}{2}}(n-1) = t_{0.025}(49) \approx 2.01$。因为 $|t| = 1.397 < 2.01$,所以接受原假设 $H_0: \mu = 3$。

3. 单个正态总体的方差检验

设 $X_1, X_2, \cdots, X_n$ 是从正态总体 $N(\mu, \sigma^2)$ 中抽取的样本,欲假设检验 $H_0: \sigma^2 = \sigma_0^2$, $H_1: \sigma^2 \neq \sigma_0^2$

若 H_0 成立,则

$$\chi^2 = \frac{\sum_{i=1}^{n}(X_i - \overline{X})^2}{\sigma_0^2}$$

服从$\chi^2(n-1)$分布。在给定显著性水平 α 的情况下，查$\chi^2(n-1)$分布的上侧分位数表得到临界值$\chi^2_{1-\frac{\alpha}{2}}$和$\chi^2_{\frac{\alpha}{2}}$，使

$$P(\chi^2 < \chi^2_{1-\frac{\alpha}{2}}) = \frac{\alpha}{2} \quad 和 \quad P(\chi^2 > \chi^2_{\frac{\alpha}{2}}) = \frac{\alpha}{2}$$

检验的拒绝域为

$$W = \{\chi^2 < \chi^2_{1-\frac{\alpha}{2}} 或 \chi^2 > \chi^2_{\frac{\alpha}{2}}\}$$

由样本 $X_1, X_2, \cdots, X_n$ 的观测值计算χ^2值，若$\chi^2 < \chi^2_{1-\frac{\alpha}{2}}$或$\chi^2 > \chi^2_{\frac{\alpha}{2}}$，则拒绝原假设 H_0，否则接受原假设 H_0。

这种检验法被称为χ^2检验法。

例 7.3.3　已知某车间生产金属丝，质量向来比较稳定，折断力方差 $\sigma_0^2 = 64$。今从一批产品中抽出 10 根做折断力试验，结果分别为(单位：kg)：

578　572　570　568　572　570　572　596　584　570

则在显著性水平 $\alpha = 0.05$ 下，是否可以认为该批金属丝的折断力方差也是 64?

设总体 X 表示该车间生产的金属丝折断力，则 X 服从正态分布 $N(\mu, \sigma^2)$，在这里需要检验

$$H_0: \sigma^2 = 64 \quad H_1: \sigma^2 \neq 64$$

使用χ^2检验法。由样本数据可以得到

$$\bar{x} = (578 + \cdots + 570)/10 = 575.2$$

$$\sum_{i=1}^{10}(x_i - \bar{x})^2 = (578 - 575.2)^2 + \cdots + (570 - 575.2)^2 = 681.6$$

因而得到

$$\chi^2 = \frac{\sum_{i=1}^{10}(x_i - \bar{x})^2}{\sigma_0^2} = \frac{681.6}{64} = 10.65$$

在显著性水平 $\alpha = 0.05$，自由度为 $n-1=9$ 的情况下，查χ^2分布的分位数表得到 $\chi^2_{1-\frac{\alpha}{2}} = 2.7$，$\chi^2_{\frac{\alpha}{2}} = 19.023$。因为$\chi^2_{1-\frac{\alpha}{2}} < \chi^2 < \chi^2_{\frac{\alpha}{2}}$，所以接受原假设 H_0。

7.3.3　两个正态总体的假设检验

1. 已知方差时的两个正态总体的均值检验

设总体 $X \sim N(\mu_1, \sigma_1^2)$，$Y \sim N(\mu_2, \sigma_2^2)$，方差 σ_1^2 与 σ_2^2 为已知常数，且 X 与 Y 相互独立，令

$$H_0: \mu_1 = \mu_2 \quad H_1: \mu_1 \neq \mu_2$$

现独立地分别从两个总体 X 和 Y 中抽取样本 $X_1, X_2, \cdots, X_m$ 和 $Y_1, Y_2, \cdots, Y_n$，记

$$\overline{X} = \frac{1}{m}\sum_{i=1}^{m} X_i,\quad \overline{Y} = \frac{1}{n}\sum_{i=1}^{n} Y_i$$

$$S_1^2 = \frac{1}{m-1}\sum_{i=1}^{m}(X_i - \overline{X})^2,\quad S_2^2 = \frac{1}{n-1}\sum_{i=1}^{n}(Y_i - \overline{Y})^2$$

当 H_0 成立时，统计量

$$U = \frac{\overline{X} - \overline{Y}}{\sqrt{\frac{\sigma_1^2}{m} + \frac{\sigma_2^2}{n}}}$$

服从标准正态分布 $N(0,1)$。在给定显著性水平 α 的情况下，查正态分布表计算 $u_{\frac{\alpha}{2}}$，使得

$$P\{\,|U| > \mu_{\frac{\alpha}{2}}\} = \alpha$$

从而检验的拒绝域为

$$W = \{\,|U| > \mu_{\frac{\alpha}{2}}\}$$

由样本 $X_1, X_2, \cdots, X_n$ 的观测值计算 U 的观测值 μ，若 $|\mu| > \mu_{\frac{\alpha}{2}}$，则拒绝原假设 H_0，否则接受原假设 H_0。

2. 方差未知但相等时两个正态总体的均值检验

设总体 $X \sim N(\mu_1, \sigma_1^2)$，$Y \sim N(\mu_2, \sigma_2^2)$，$\sigma_1^2$ 与 σ_2^2 均为未知参数但已知 $\sigma_1^2 = \sigma_2^2$，令

$$H_0: \mu_1 = \mu_2 \quad H_1: \mu_1 \neq \mu_2$$

现独立地分别从两个总体 X 和 Y 中抽取样本 $X_1, X_2, \cdots, X_m$ 和 $Y_1, Y_2, \cdots, Y_n$，记

$$\overline{X} = \frac{1}{m}\sum_{i=1}^{m} X_i,\quad \overline{Y} = \frac{1}{n}\sum_{i=1}^{n} Y_i$$

$$S_1^2 = \frac{1}{m-1}\sum_{i=1}^{m}(X_i - \overline{X})^2,\quad S_2^2 = \frac{1}{n-1}\sum_{i=1}^{n}(Y_i - \overline{Y})^2$$

当 H_0 成立时，统计量

$$T = \frac{\overline{X} - \overline{Y}}{\sqrt{(m-1)S_1^2 + (n-1)S_2^2}}\sqrt{\frac{mn(m+n-2)}{m+n}}$$

服从自由度为 $(m+n-2)$ 的 t 分布。在给定显著性水平 α 的情况下，查 t 分布表计算 $t_{\frac{\alpha}{2}}$，使

$$P\{\,|T| > t_{\frac{\alpha}{2}}\} = \alpha$$

从而检验的拒绝域为

$$W = \{\,|T| > t_{\frac{\alpha}{2}}\}$$

由样本 $X_1, X_2, \cdots, X_n$ 的观测值计算 T 的观测值 t，若 $|t| > t_{\frac{\alpha}{2}}$，则拒绝原假设 H_0，否则接受原假设 H_0。

例 7.3.4 设有甲、乙两台机床，加工同样产品，从这两台机床加工的产品中随机抽取若干产品，测得产品直径如下。（单位：mm）

甲：20.5 19.8 19.7 20.4 20.1 20.0 19.6 19.9

乙：19.7 20.8 20.5 19.8 19.4 20.6 19.2

已知方差相同，试比较在显著性水平 $\alpha=0.05$ 下，甲、乙两台机床的加工精度有无显著差异？

设总体 X 表示甲机床加工的产品直径，总体 Y 表示乙机床加工的产品直径，则 $X\sim N(\mu_1,\sigma_1^2)$，$Y\sim N(\mu_2,\sigma_2^2)$，$\sigma_1^2$ 与 σ_2^2 均为未知参数但 $\sigma_1^2=\sigma_2^2$，在这里需要检验

$$H_0: \mu_1=\mu_2 \quad H_1: \mu_1\neq\mu_2$$

使用 t 检验法。由样本数据得

$$\bar{x}=(20.5+19.8+19.7+20.4+20.1+20.0+19.6+19.9)/8=20$$

$$\bar{y}=(19.7+20.8+20.5+19.8+19.4+20.6+19.2)/7=20$$

$$(m-1)S_1^2=\sum_{i=1}^{m}(x_i-\bar{x})^2=(20.5-20)^2+\cdots+(19.9-20)^2=0.72$$

$$(n-1)S_2^2=\sum_{i=1}^{m}(y_i-\bar{y})^2=(19.7-20)^2+\cdots+(19.2-20)^2=2.38$$

$$t=\frac{\bar{x}-\bar{y}}{\sqrt{(m-1)S_1^2+(n-1)S_2^2}}\sqrt{\frac{mn(m+n-2)}{m+n}}=0$$

在显著性水平 $\alpha=0.05$，自由度为 $(m+n-2)=8+7-2=13$ 下，查 t 分位表得临界值 $t_{\frac{\alpha}{2}}=t_{0.025}(13)=2.16$。因为 $|t|<t_{\frac{\alpha}{2}}$，故接受原假设，即甲、乙两台机床的加工精度无显著差异。

3. 两个正态总体的方差检验

设总体 $X\sim N(\mu_1,\sigma_1^2)$，$Y\sim N(\mu_2,\sigma_2^2)$，且相互独立，令

$$H_0:\sigma_1^2=\sigma_2^2 \quad H_1:\sigma_1^2\neq\sigma_2^2$$

现独立地分别从两个总体 X 和 Y 中抽取样本 $X_1,X_2,\cdots,X_m$ 和 $Y_1,Y_2,\cdots,Y_n$，记

$$\overline{X}=\frac{1}{m}\sum_{i=1}^{m}X_i,\quad \overline{Y}=\frac{1}{n}\sum_{i=1}^{n}Y_i$$

$$S_1^2=\frac{1}{m-1}\sum_{i=1}^{m}(X_i-\overline{X})^2,$$

$$S_2^2=\frac{1}{n-1}\sum_{i=1}^{n}(Y_i-\overline{Y})^2$$

当 H_0 成立时，统计量

$$F=\frac{S_1^2}{S_2^2}$$

服从自由度为 $m-1$ 和 $n-1$ 的 F 分布。在给定显著性水平 α 的情况下，查 F 分布表计算 $F_{1-\frac{\alpha}{2}}$ 和 $F_{\frac{\alpha}{2}}$，使

$$P(F<F_{1-\frac{\alpha}{2}})=\frac{\alpha}{2} \quad 和 \quad P(F>F_{\frac{\alpha}{2}})=\frac{\alpha}{2}$$

检验的拒绝域为

$$W=\{F<F_{1-\frac{\alpha}{2}} \quad 或 \quad F>F_{\frac{\alpha}{2}}\}$$

由样本 $X_1,X_2,\cdots,X_n$ 的观测值计算 F 值，若 $F<F_{1-\frac{\alpha}{2}}$ 或 $F>F_{\frac{\alpha}{2}}$，则拒绝原假设 H_0，否则接受原假设 H_0。

不难看出，$1/F$ 仍然服从 F 分布。在实际应用中，首先计算出 S_1^2 和 S_2^2，然后按大小分别记为 $S_{大}^2$ 和 $S_{小}^2$，再按 $F=S_{大}^2/S_{小}^2$ 算出 F 的观测值，它只需与临界值 $F_{\frac{\alpha}{2}}(n_1-1,n_2-1)$ 进行比较，n_1 和 n_2 分别是对应于分子和分母的样本容量。若 $F>F_{\frac{\alpha}{2}}(n_1-1,n_2-1)$，则拒绝原假设 H_0，否则接受原假设 H_0。

例 7.3.5 设甲、乙两车间生产的电灯泡的寿命都服从正态分布，从甲、乙车间分别抽取 50 个、60 个样品，测得其寿命数据如下。

甲车间：$m=50$ 个，$\bar{x}=1282\text{h}$，$S_1=80\text{h}$

乙车间：$n=60$ 个，$\bar{y}=1208\text{h}$，$S_2=94\text{h}$

试求这两个车间生产的灯泡寿命的方差是否相同？

设总体 X 表示甲车间生产的电灯泡的寿命，总体 Y 表示乙车间生产的电灯泡的寿命，则 $X\sim N(\mu_1,\sigma_1^2)$，$Y\sim N(\mu_2,\sigma_2^2)$，在这里需要检验

$$H_0:\sigma_1^2=\sigma_2^2 \quad H_1:\sigma_1^2\neq\sigma_2^2$$

使用 t 检验法。由样本数据得

$$S_{大}=94,\quad S_{小}=80,\quad n_1=60,\quad n_2=50$$

所以

$$F=\frac{S_{大}^2}{S_{小}^2}=\frac{94^2}{80^2}=1.38$$

对于显著性水平 $\alpha=0.05$，自由度为 $n_1-1=59$ 和 $n_2-1=49$，查 F 分布分位数表得 $F_{\frac{\alpha}{2}}(59,49)=1.8$。因为 $F<F_{\frac{\alpha}{2}}(59,49)$，所以接受原假设。

7.3.4 χ^2 拟合检验

上面介绍的几种检验都是在总体分布形式已知的情况下进行的。在实际问题中，常常会出现总体分布未知，需要根据样本来检验关于分布的假设。这里介绍一种比较常用的 χ^2 拟合检验法。

设总体 X 分布未知，令

H_0：总体 X 分布函数为 $F(X)$

H_1：总体 X 分布函数不是 $F(X)$

将结果总体 Ω 分为 k 个互不相容的事件 $A_1,A_2,\cdots,A_k$，其中 $\sum_{i=1}^{k}A_i=\Omega$，$i\neq j$，$i,j=1,2,\cdots,k$。在假设 H_0 下，计算 $p_i=P(A_i)$，$i=1,2,\cdots,k$。现进行 n 次独立的试验，设事件 A_i 出现的频数为 f_i，则频率为 f_i/n。若 H_0 为真，f_i/n 与 p_i 的差异应该不大。

现构造统计量

$$\chi^2=\sum_{i=1}^{k}\frac{(f_i-np_i)^2}{np_i}$$

当 n 充分大时，若 H_0 为真，χ^2 近似地服从自由度为 $k-r-1$ 的 χ^2 分布，其中 r 是被估计的参数个数。

由样本 $X_1,X_2,\cdots,X_n$ 的观测值计算 χ^2 值，若在显著性水平 α 的情况下 $\chi^2\geqslant\chi^2_\alpha$

$(k-r-1)$，则拒绝 H_0，否则就接受 H_0。

以上的检验法也称为皮尔逊χ^2拟合检验，该检验法要求 n 要足够大，以及 np_i 不太小。根据实践的经验，要求样本容量 $n\geqslant 50$，以及每一个 np_i 都在 5 以上，否则适当地合并 A_i，以满足这个要求。

例 7.3.6　从随机数表抽取 200 个观察数据，经整理列入表 7.1 中。

表 7.1　例 7.3.6 用表

分组	0～	0.1～	0.2～	0.3～	0.4～	0.5～	0.6～	0.7～	0.8～	0.9～
频数 f_i	23	21	26	17	15	15	25	14	25	19

试在显著性水平为 $\alpha=0.05$ 的情况下，检验它是否服从[0,1]上的均匀分布？

在这里需要假设检验

H_0：随机数服从[0,1]上的均匀分布

H_1：随机数不服从[0,1]上的均匀分布

当 H_0 为真时，$p_i=0.1$，$np_i=0.1\times 200=20$。因此

$$\chi^2=\sum_{i=1}^{10}\frac{(f_i-np_i)^2}{np_i}=9.6$$

由χ^2分布表查得$\chi^2_{0.05}(9)=16.92$。因为 $9.6<16.96$，所以接受 H_0。

7.4　应用举例

数理统计知识在信息安全领域有着广泛而深入的应用，如密码学、数字水印及网络入侵检测等。限于篇幅，这里只简要介绍两个关于假设检验在密码检测中的应用实例。

7.4.1　频数检测

一个安全的密码算法可看作是一个随机数发生器，它所产生的序列应该满足各种随机特性，即能够通过各项随机性检测。频数检测是最基本的随机性检测方法，用来检测一个序列中 0 和 1 的个数是否接近。在进行随机性检测时，应该首先选择进行该项检测，只有该项检测通过后再选择继续进行其他检测。

根据抛币模型，理想的随机序列的产生可看成是投掷硬币的结果，即根据硬币正、反面标记为“0”或“1”，对于每一次投掷结果，“0”或“1”出现的概率均为 1/2，并且，投掷结果之间相互独立。因此，一个随机的二元序列的每一位都应该服从二点分布，并且 0 和 1 出现的概率都为 1/2。令 $\varepsilon_1\varepsilon_2\cdots\varepsilon_n\,(\varepsilon_i\in\{0,1\},1\leqslant i\leqslant n)$表示待检序列，构造变量 $X_i=2\varepsilon_i-1(1\leqslant i\leqslant n)$。记 $S_n=X_1+X_2+\cdots+X_n=2(\varepsilon_1+\varepsilon_2+\cdots+\varepsilon_n)-n$，根据中心极限定理，有

$$\frac{\sum X_i-\sum E(X_i)}{\sqrt{\sum D(X_i)}}\to N(0,1)$$

$$E(X_i) = -1 \times 1/2 + 1 \times 1/2 = 0$$
$$D(X_i) = E(X_i^2) - E(X_i)^2 = 1$$

所以，有

$$\phi(z) = \lim_{n\to\infty} P\left(\frac{S_n}{\sqrt{n}} \leqslant z\right) \equiv \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{z} e^{-\frac{\mu^2}{2}} d\mu$$

对于非负数 z，有

$$P\left(\frac{|S_n|}{\sqrt{n}} \leqslant z\right) = 2\phi(z) - 1$$

因此，统计值 $V = \frac{S_n}{\sqrt{n}} = \frac{X_1 + X_2 + \cdots + X_n}{\sqrt{n}}$ 应服从标准正态分布 $N(0,1)$。

下面介绍一下具体的检测过程。

给定 $\varepsilon_1\varepsilon_2\cdots\varepsilon_n(\varepsilon_i \in \{0,1\}, 1 \leqslant i \leqslant n)$，若想检验该序列是否由理想的随机数发生器产生（即为原假设 H_0），则首先构造统计值 $V = \frac{S_n}{\sqrt{n}}$。

然后在给定显著性水平 α 的情况下，查正态分布表计算 $z_{\frac{\alpha}{2}}$，使

$$P\{|V| > z_{\frac{\alpha}{2}}\} = \alpha$$

从而检验的拒绝域为

$$W = \{|V| > z_{\frac{\alpha}{2}}\}$$

根据样本观察值计算统计量的具体值 V，若落入拒绝域中，则在显著性水平 α 条件下拒绝原假设 H_0，即认为被测序列不随机；否则就接受原假设 H_0。

7.4.2 分组密码明密文独立性检测

分组密码明密文独立性检测主要是检测密文是否有不依赖于明文统计特性的性质，即对于任意给定的明文，在分组密码的作用下得到密文，则明文与其对应的密文的距离应是随机的。即每一位密文与其对应的明文比特相等的概率为1/2。

设待测分组密码算法的分组长度为 n 比特，随机生成 F 个明文分组 $P_0, P_1, P_2, \cdots, P_{F-1}$ 和1个密钥。在ECB模式下使用该密钥逐个加密明文分组，得到 F 个密文分组 $C_0, C_1, C_2, \cdots, C_{F-1}$。记录相应的明密文距离 $D_i = W(P_i \oplus C_i), 0 \leqslant i \leqslant F-1$，$W$ 表示计算汉明重量（比特串中1的个数）的函数。统计 $D_i(0 \leqslant i \leqslant F-1)$ 为 $w(0 \leqslant w \leqslant n)$ 的分组数，记为 H_w。

对于一个好的分组密码，H_w 应该符合二项分布 $B(n,1/2)$。在这里使用皮尔逊 χ^2 拟合检验在判断 H_w 是否符合二项分布 $B(n,1/2)$。因为 H_w 的期望数 $E_w = C_n^w \times F/2^n$，所以需要计算 $\chi^2 = \sum_{i=0}^{n} \frac{(H_i - E_i)^2}{E_i}$，将 χ^2 计算结果与显著性水平为 α、自由度为 n 的 χ^2 阈值相比较，若 $\chi^2 < \chi^2{}_{\alpha}(n)$，则认为 H_w 符合二项分布 $B(n,1/2)$，即待测分组密码算法满足明密文独立性特性，否则，认为待测分组密码算法不满足明密文独立性特性。

因为皮尔逊 χ^2 拟合检验要求 E_w 不能太小，所以在实际检测中，常常需要将 E_w

进行合并，设 k 是实际的分组数，则将最后计算的 χ^2 值与显著性水平为 α、自由度为 $k-1$ 的 χ^2 阈值相比较，若 $\chi^2 < \chi^2_\alpha(k-1)$，则认为 H_w 符合二项分布 $B(n,1/2)$。

7.5 注记

本章重点介绍了一些在信息安全研究中常用的数理统计知识和方法，同时用典型实例阐述了数理统计在信息安全领域中的应用。本章所介绍的数理统计知识及例子主要来自于文献[1]～[6]。其中关于 7.1.3 小节中常用统计量分布的详细证明过程可以参看文献[1]。若读者想了解一下除本章以外的更深入的数理统计知识，可以仔细研读文献[5]、[6]。另外，本章所举的应用实例来自于文献[7]、[8]。

参 考 文 献

[1] 王成名，余鑫晖. 应用概率统计. 桂林：广西师范大学出版社，1994
[2] 盛骤，谢式千，潘承毅. 概率论与数理统计. 第二版. 北京：高等教育出版社，1997
[3] 陈希孺. 概率论与数理统计. 北京：科学出版社，2000
[4] 王福保. 概率论及数理统计. 第三版. 上海：同济大学出版社，1993
[5] 茆诗松，王静龙，濮晓龙. 高等数理统计. 北京：高等教育出版社，施普林格出版社，1998
[6] Hogg R V，Craig A T. 数理统计学导论. 第 5 版. 北京：高等教育出版社，2004
[7] Rukhin A，et. al. A Statistical Test Suite for the Validation of Random and Pseudo Random Number Generators for Cryptographic Applications. NIST Special Publication 800-22 (revised May 15，2001)
[8] 冯登国，吴文玲. 分组密码的分析与设计. 北京：清华大学出版社，2000

第 8 章　随机过程方法与技术

随机过程是对一连串随机事件间动态关系的定量描述。它是自然科学、工程科学、社会科学各领域研究随机现象的有力工具。本章主要介绍随机过程的概念和统计描述方法，几个从实际问题抽象出的著名随机过程及其统计特性，并以马尔柯夫密码为例介绍随机过程方法与技术在密码学中的应用。

8.1　随机过程的概念和记号

随机过程被认为是概率论的“动力学”部分，意思是说，它的研究对象是随时间演变的随机现象。用数学语言来说，就是事物变化的过程不能用一个(或几个)时间 t 的确定函数加以描述。或者从另一个角度来看，对事物变化的全过程进行一次观察得到的结果是一个时间 t 的函数，但对同一事物的变化过程独立地重复进行多次观察所得的结果是不相同的，而且每次观察之前不能预知试验结果。现在来看一个具体例子。

例 8.1.1(热噪声问题)　考虑电子网络中的一个电阻，由于电阻中自由电子的随机运动，导致电阻两端的电压有一个随机的起伏，这一起伏的电压就称为热噪声。在无线电通信技术中，接收机在接收信号时，机内的热噪声要对信号产生持续的干扰，为消除这种干扰，就必须考虑热噪声随时间变化的过程。

为了考察热噪声随时间变化的过程，通过某种装置对电阻两端的热噪声电压进行长时间的测量，并把结果自动记录下来，这作为一次试验结果，便得到一个电压-时间函数(即电压关于时间 t 的函数)$u_1(t)$，如图 8.1 所示，这个电压-时间函数是不可能预先确知的，只有通过测量才能得到。如在相同条件下独立地再进行一次测量，则得到的记录是不同的。事实上，由于热噪动的随机性，在相同条件每次测量都将产生不同的电压-时间函数(图 8.1 可以设想为在相同条件下同时对无限多个“相同”电阻作测量的结果)。

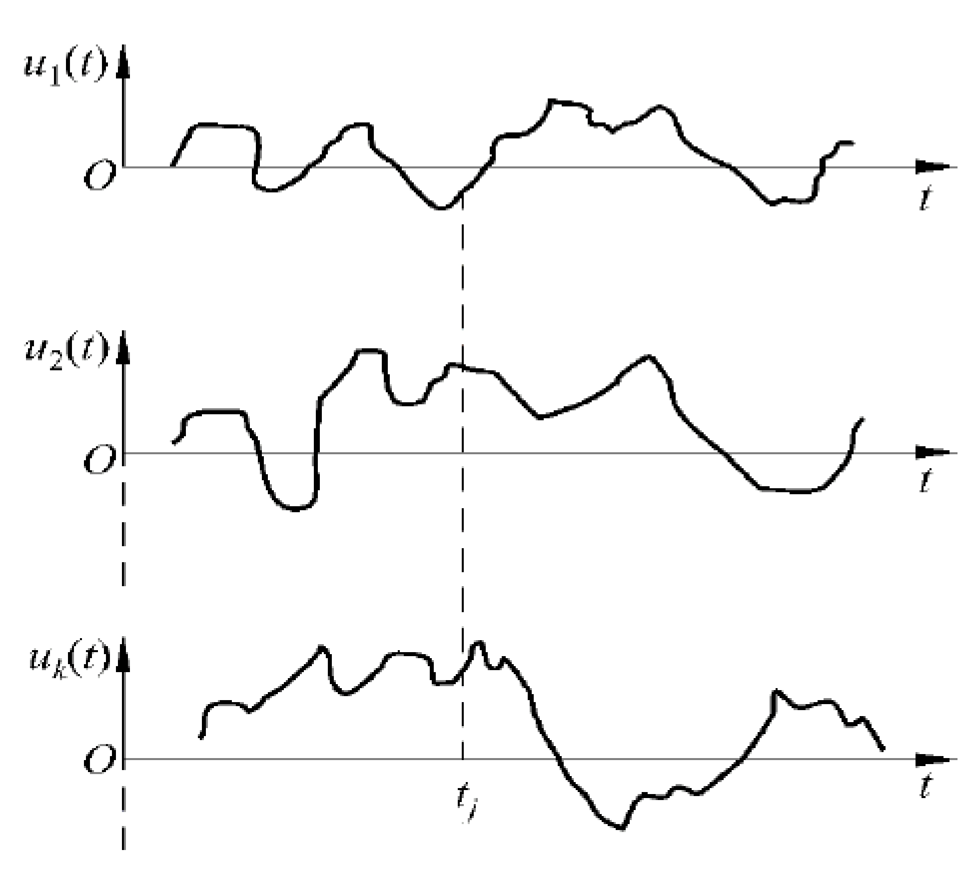


图 8.1　电压-时间特性

如此，也可以把对电阻热噪声电压的变化过程的观察看作一个随机试验，只是这里，每次试验需在某个时间范围内持续进行，而相应的试验结果则是一个时间 t 的函数。随机试验可以用其所有可能的试验结果所构成的样本空间来描述。同样可以用电阻所可能产生的一族电压-时间函数，即用$\{u_1(t)\}$来描述其热噪声电压的变化过程。

现以上述例子为背景，引入随机过程的概念。

设 E 是随机试验，$S=\{e\}$ 是它的样本空间。如果对于每一个 $e\in S$，总可以依某种规则确定一参数为 t 的实值函数

$$X(e,t),\quad t\in T$$

与之对应，于是，当 e 取遍 S 时，就得到定义在 T 上的一族函数，称此族参数 t 的函数为**随机过程**，而族中每一个函数称为这个随机过程的**样本函数**。T 是参数 t 的变化范围，称为**参数集**。T 一般表示时间集合。

依照上述说法，热噪声电压的变化过程是一随机过程，一次观察得到的电压-时间函数就是这个随机过程的一个样本函数。

随机过程也可以看成是两个变量 e 和 t 的函数：$X(e,t)$，$e\in S$，$t\in T$。

$X(e,t)$ 的含义是：

(1) 对于一个特定的试验结果 $e_i\in S$，$X(e_i,t)$ 就是对应于 e_i 的样本函数，简记为 $x_i(t)$，它可以理解为随机过程的**一次实现**。

(2) 对于每一个固定的参数 $t_j\in T$，$X(e,t_j)$ 是一个定义在 S 上的随机变量(参见图 8.1)。工程上有时把 $X(e,t_j)$ 称为随机过程在 $t=t_j$ 时的**状态**。而 $X(e,t_j)=x$ 说成是 $t=t_j$ 时，随机过程处于状态 x。对于一切 $e\in S$，$t\in T$，$X(e,t)$ 所能取的一切值的集合，称为随机过程的**状态空间**。

依照(2)的含义，可给出随机过程另一种常用的描述方式。

随机过程是(定义在 S 上的)依赖于参数 $t\in T$ 的一族随机变量，并记为 $\{X(e,t),t\in T\}$。

在以后的叙述中，为简便起见，省去随机过程记号中的 e，以 $\{X(t),t\in T\}$ 或 $\{X_t,t\in T\}$ 或 $X(t),t\in T$ 表示随机过程。在上下文不致混淆的情形下，一般略去记法中的参数集 T。

随机过程的不同描述方式本质上是一致的。在理论分析时往往以随机变量族的描述方式，而在实际测量和处理中往往采用样本函数族的描述方式，这两种描述方式在理论和实际两方面是互为补充的。

例 8.1.2　设一个电话交换台迟早会接到用户的呼叫，以 $X(t)$ 表示时间间隔 $[0,t)$ 内交换台接到的呼叫次数，它是一个随机变量，且对于不同的 $t\geqslant 0$，$X(t)$ 是不同的随机变量。于是，$\{X(t),t\geqslant 0\}$ 是一随机过程。

例 8.1.3　考虑抛掷一颗骰子的实验。

(1) 设 X_n 是第 n 次($n\geqslant 1$)抛掷的点数，对于 $n=1,2,\cdots$ 的不同值，X_n 是不同的随机变量，因而 $\{X_n,n\geqslant 1\}$ 构成一随机过程，称为**伯努利过程**或**伯努利随机序列**。

(2) 设 X_n 是前 n 次抛掷中出现的最大点数，$\{X_n,n\geqslant 1\}$ 也是一随机过程。

随机过程可依其在任一时刻的状态是连续型随机变量或离散型随机变量而分成**连续型随机过程**或**离散型随机过程**。热噪声电压是连续型随机过程，例 8.1.2 和例 8.1.3 是离散型随机过程。

随机过程还可依时间(参数)是连续或离散进行分类。当时间集 T 是有限或无限区间时，称 $\{X(t),t\in T\}$ 为**连续参数随机过程**。如果 T 是离散集合，如 $T=\{0,1,$

$2,\cdots\}$或$\{0,\pm1,\pm2,\cdots\}$,则称$\{X(t),t\in T\}$为**离散参数随机过程**或**随机序列**,此时常记成$\{X_n,n=0,1,2,\cdots\}$等。

最后指出,参数t虽然通常解释为时间,但它也可以表示其他的量,诸如序号、距离等。例如,在例8.1.3中,假定每隔一个单位时间抛掷骰子一次,那么第n次抛掷时骰子出现的点数X_n就相当于$t=n$时骰子出现的点数。

8.2 随机过程的统计描述

随机过程在任一时刻的状态是随机变量,由此可以利用随机变量的统计描述方法来描述随机过程的统计特性。

8.2.1 随机过程的分布函数族

给定随机过程$\{X(t),t\in T\}$。对于每一个固定的$t\in T$,随机变量$X(t)$的分布函数一般与t有关,记为

$$F(x,t)=P\{X(t)\leqslant x\},\quad x\in R$$

称它为随机过程$\{X(t),t\in T\}$的**一维分布函数**,而$\{F(x,t),t\in T\}$称为**一维分布函数族**。

一维分布函数族刻画了随机过程在各个个别时刻的统计特性。为了描述随机过程在不同时刻状态之间的统计联系,一般可对任意$n(n=2,3,\cdots)$个不同的时刻$t_1,t_2,\cdots,t_n\in T$,引入n维随机变量$(X(t_1),X(t_2),\cdots,X(t_n))$,它的分布函数记为

$$F(x_1,x_2,\cdots,x_n;t_1,t_2,\cdots,t_n)=P\{X(t_1)\leqslant x_1,X(t_2)\leqslant x_2,\cdots,X(t_n)\leqslant x_n\}$$

$$x_i\in R,\quad i=1,2,\cdots,n$$

对于固定的n,则称$\{F(x_1,x_2,\cdots,x_n;t_1,t_2,\cdots,t_n),t_i\in T\}$为随机过程$\{X(t),t\in T\}$的**$n$维分布函数族**。

当n充分大时,n维分布函数族能够近似地描述随机过程的统计特性。显然,n取得越大,则n维分布函数族描述随机过程的特性也越完善。一般可以说,**有限维分布函数族**,即$\{F(x_1,x_2,\cdots,x_n;t_1,t_2,\cdots,t_n),n=1,2,\cdots,t_i\in T\}$,完全地确定了随机过程的统计特性。

在上一节曾将随机过程按其状态或时间的连续或离散进行了分类。然而,随机过程的本质的分类方法乃是按其分布特性进行分类。具体地说,就是依照过程在不同时刻的状态之间的特殊统计依赖方式,抽象出一些不同类型的模型,如马尔柯夫过程、平稳过程等。将在以后的几节对它们做不同程度的介绍。

8.2.2 随机过程的数字特征

随机过程的分布函数族能完善地刻画随机过程的统计特性,但是在实际中根据观察往往只能得到随机过程的部分资料(样本),用它们来确定有限维分布函数族是困难的,甚至是不可能的。因而有必要引入随机过程的基本的数字特征——均值函数和相关函数等。下面将会看到,这些数字特征在一定条件下是便于测量的。

给定随机过程$\{X(t),t\in T\}$。固定$t\in T$，$X(t)$是一随机变量，它的均值一般与t有关，记为

$$\mu_X(t)=E[X(t)] \tag{8.1}$$

则称$\mu_X(t)$为随机过程$\{X(t),t\in T\}$的**均值函数**。

注意，$u_X(t)$是随机过程的所有样本函数在时刻t的函数值的平均值，通常称这种平均为集平均或统计平均。

其次，把随机变量$X(t)$的二阶原点矩和二阶中心矩分别记做

$$\psi_X^2(t)=E[X^2(t)] \tag{8.2}$$

$$\sigma_X^2(t)=D_X(t)=\mathrm{Var}[X(t)]=E\{[X(t)-\mu_X(t)]^2\} \tag{8.3}$$

分别称它们为随机过程$\{X(t),t\in T\}$的**均方值函数**和**方差函数**。方差函数的算术根$\sigma_X(t)$称为随机过程的**均方差函数**，它表示随机过程$X(t)$在时刻t对于均值$\mu_X(t)$的平均偏离程度。

又设任意$t_1,t_2\in T$，把随机变量$X(t_1)$和$X(t_2)$的二阶原点混合矩记做

$$R_{XX}(t_1,t_2)=E[X(t_1)X(t_2)] \tag{8.4}$$

并称它为随机过程$\{X(t),t\in T\}$的**自相关函数**，简称**相关函数**。

类似地，还可写出$X(t_1)$和$X(t_2)$的二阶中心混合矩

$$C_{XX}(t_1,t_2)=E\{[X(t_1)-\mu_X(t_1)][X(t_2)-\mu_X(t_2)]\} \tag{8.5}$$

并称它为随机过程$\{X(t),t\in T\}$的**自协方差函数**，简称**协方差函数**。

自相关函数和自协方差函数是刻画随机过程自身在两个不同时刻的状态之间统计依赖关系的数字特征。现把式(8.1)至式(8.5)定义的诸数字特征之间的关系简述如下：

由式(8.2)和式(8.4)知

$$\Psi_X^2(t)=R_{XX}[t,t] \tag{8.6}$$

由式(8.5)展开，得

$$C_{XX}(t_1,t_2)=R_{XX}(t_1,t_2)-\mu_X(t_1)\mu_X(t_2) \tag{8.7}$$

特别，当$t_1=t_2=t$时，由式(8.7)，得

$$\sigma_X^2(t)=C_{XX}(t,t)=R_{XX}(t,t)-\mu_X^2(t) \tag{8.8}$$

由式(8.6)至式(8.8)可知，以上诸数字特征中最主要的是均值函数和自相关函数。从理论角度来看，仅仅研究均值函数和自相关函数当然是不能代替对整个随机过程的研究的，但是由于它们确实刻画了随机过程的主要统计特性，而且较有限维分布函数族易于观察和实际计算，因而对实际应用而言，它们常常能够起到重要的作用。据此，在随机过程理论中着重研究所谓二阶矩过程。

随机过程$\{X(t),t\in T\}$，如果对每一个$t\in T$，二阶矩$E[X^2(t)]$都存在，那么称它为**二阶矩过程**。

二阶矩过程的相关函数总存在。事实上，由于$E[X^2(t_1)]$和$E[X^2(t_2)]$都存在，依据柯西-许瓦兹不等式有

$$\{E[X(t_1)X(t_2)]\}^2\leqslant E[X^2(t_1)]E[X^2(t_2)],\quad t_1,t_2\in T$$

即知$R_{XX}[t_1,t_2]=E[X(t_1)X(t_2)]$存在。

例 8.2.1 设 A、B 是两个随机变量，试求随机过程 $X(t)=At+B, t\in T=(-\infty,+\infty)$ 的均值函数和自相关函数。如果 A、B 相互独立，且 $A\sim N(0,1)$，B 在区间(0,2)上均匀分布，问 $X(t)$ 的均值函数和自相关函数是什么？

解：$X(t)$ 的均值函数和自相关函数分别为

$$\mu_X(t) = E[At+B] = tE[A]+E[B]$$

$$\begin{aligned} R_{XX}(t_1,t_2) &= E[(At_1+B)(At_2+B)] \\ &= t_1t_2E[A^2]+(t_1+t_2)E[AB]+E[B^2] \end{aligned}$$

当 $A\sim N(0,1)$ 时，$E(A)=0$，$E(A^2)=1$；当 B 在区间(0,2)上均匀分布时，$E(B)=1$，$E(B^2)=\frac{4}{3}$；又因为 A、B 独立，故 $E[AB]=E[A]E[B]=0$。所以，此时 $\mu_X(t)=1$，$R_{XX}(t_1,t_2)=t_1t_2+\frac{4}{3}$。

8.2.3 二维随机过程的分布函数和数字特征

在实际中，有时需同时研究两个或两个以上随机过程及其之间的统计关系。例如，某地在时段$[0,t)$内的最高温度 $X(t)$ 和最低温度 $Y(t)$ 都是随机过程，需要研究它们的统计关系。又如，输入到一个系统的信号和噪声可以都是随机过程，这时输出也是随机过程，需要研究输出与输入之间的统计关系。对于这类问题，除了对各个随机过程的统计特性加以研究外，还必须将几个随机过程作为整体研究其统计特性。

设 $X(t)$、$Y(t)$ 是定义在同一样本空间 S 和统一参数集 T 上的随机过程，对于不同的 $t\in T$，$(X(t),Y(t))$ 是不同的二维随机变量，则称 $\{(X(t),Y(t)),t\in T\}$ 为**二维随机过程**。

给定二维随机过程 $\{(X(t),Y(t)),t\in T\}$，$t_1,t_2,\cdots,t_n;t_1',t_2',\cdots,t_m'$ 是 T 中任意两组实数，则称 $n+m$ 维随机变量

$$(X(t_1),X(t_2),\cdots,X(t_n);\quad Y(t_1'),Y(t_2'),\cdots,Y(t_m'))$$

的分布函数

$$F(x_1,x_2,\cdots,x_n;t_1,t_2,\cdots,t_n;y_1,y_2,\cdots,y_m;t_1',t_2',\cdots,t_m')$$

$$x_i,y_j\in R, i=1,2,\cdots,n, j=1,2,\cdots,m$$

为这个二维随机过程的 **$n+m$ 维分布函数**或随机过程 $X(t)$ 与 $Y(t)$ 的 **$n+m$ 维联合分布函数**。同样可以定义二维随机过程的 $n+m$ 维分布函数族和有限维分布函数族。

如果对于任意的正整数 n、m，任意的数组 $t_1,t_2,\cdots,t_n\in T$，$t_1',t_2',\cdots,t_m'\in T$，$x_1,x_2,\cdots,x_n\in R$，$y_1,y_2,\cdots,y_m\in R$，上述 $n+m$ 维分布函数恒等于

$$F_1(x_1,x_2,\cdots,x_n;t_1,t_2,\cdots,t_n)F_2(y_1,y_2,\cdots,y_m;t_1',t_2',\cdots,t_m'),$$

其中 F_1、F_2 分别是 $X(t)$ 的 n 维分布函数和 $Y(t)$ 的 m 维分布函数，则称随机过程 $X(t)$ 与 $Y(t)$ 是**相互独立的**。

关于数字特征，除了 $X(t)$ 与 $Y(t)$ 各自的均值和自相关函数外，在应用中感兴趣的是 $X(t)$ 和 $Y(t)$ 二阶原点混合矩，记做

$$R_{XY}(t_1,t_2) = E[X(t_1)Y(t_2)],\quad t_1,t_2\in T \tag{8.9}$$

并称它为随机过程 $X(t)$ 和 $Y(t)$ 的**互相关函数**。

类似地,还有以下定义的 $X(t)$ 和 $Y(t)$ 的**互协方差函数**:

$$C_{XY}(t_1,t_2) = E\{[X(t_1)-\mu_X(t_1)][Y(t_2)-\mu_Y(t_2)]\} \quad t_1,t_2 \in T \tag{8.10}$$

如果二维随机过程 $(X(t),Y(t))$ 对任意的 $t_1,t_2\in T$ 恒有

$$C_{XY}(t_1,t_2) = 0 \tag{8.11}$$

则称随机过程 $X(t)$ 和 $Y(t)$ 是**不相关的**。

两个随机过程是相互独立的,且它们的二阶矩存在,则它们必然不相关。反之,从不相关一般并不能推断出它们是相互独立的。

8.3 泊松过程及维纳过程

泊松过程及维纳过程是两个典型的随机过程,它们在随机过程的理论和应用中都有重要的地位,它们都属于所谓独立增量过程,所以下面首先介绍独立增量过程。

给定二阶矩过程 $\{X(t),t\geqslant 0\}$,则称随机变量 $X(t)-X(s)$,$0\leqslant s<t$ 为随机过程在区间 $[s,t)$ 上的增量。如果对任意选定的正整数 n 和任意选定的 $0\leqslant t_0<t_1<\cdots<t_n$,$n$ 个增量

$$X(t_1)-X(t_0),X(t_2)-X(t_1),\cdots,X(t_n)-X(t_{n-1})$$

相互独立,则称 $\{X(t),t\geqslant 0\}$ 为**独立增量过程**。直观地说,它具有"在互不重叠的区间上,状态的增量是相互独立的"这一特征。

对于独立增量过程,可以证明:在 $X(0)=0$ 的条件下,它的有限维分布函数族可以由增量 $X(t)-X(s)$ $(0\leqslant s<t)$ 的分布所确定。

特别地,若对任意的实数 h 和 $0\leqslant s+h<t+h$,$X(t+h)-X(s+h)$ 与 $X(t)-X(s)$ 具有相同的分布,则称**增量具有平稳性**。这时,增量 $X(t)-X(s)$ 的分布函数实际上只依赖于时间差 $t-s$ $(0\leqslant s<t)$,而不依赖于 t 和 s 本身(事实上,令 $h=-s$ 即知)。当增量具有平稳性时,称相应的独立增量过程是**齐次的**。

接着,在 $X(0)=0$ 的条件下,来计算独立增量过程 $\{X(t),t\geqslant 0\}$ 的协方差函数 $C_{XX}(s,t)$。

记 $Y(t)=X(t)-\mu_X(t)$。首先注意,当 $X(t)$ 具有独立增量时,$Y(t)$ 也具有独立增量;其次,$Y(0)=0$,$E[Y(t)]=0$,且方差函数 $D_Y(t)=E[Y^2(t)]=D_X(t)$。利用这些性质,当 $0\leqslant s<t$ 时,就有

$$\begin{aligned}C_{XX}(s,t) &= E[Y(s)Y(t)] \\ &= E\{[Y(s)-Y(0)][Y(t)-Y(s)+Y(s)]\} \\ &= E[Y(s)-Y(0)]E[Y(t)-Y(s)]+E[Y^2(s)] = D_X(s)\end{aligned}$$

于是可知,对任意 $s,t\geqslant 0$,协方差函数可用方差函数表示为

$$C_{XX}(s,t) = D_X(\min(s,t)) \tag{8.12}$$

定义 8.3.1 $\{N(t),t\geqslant 0\}$ 是一状态取非负整数、时间连续的随机过程,如果它满足下述3个条件就称其为**强度为 $\lambda>0$ 的泊松过程**:

(1) $N(0)=0$;

(2) $N(t)$是独立增量过程;

(3) 对任何 $t>0,s\geqslant 0$,增量 $N(s+t)-N(t)$服从参数为 λt 的泊松分布,即

$$P\{N(s+t)-N(t)=k\}=\frac{(\lambda t)^k \exp\{-\lambda t\}}{k!},\quad k=0,1,\cdots \tag{8.13}$$

定义的条件(1)说明,随机事件从时刻 0 开始计数,条件(3)是过程称为泊松过程的直接理由。由泊松分布的性质可以得到 $E[N(t)]=\mathrm{Var}[N(t)]=\lambda t$。增量$N(s+t)-N(t)$代表时间区间$(s,s+t]$中发生的随机事件数,条件(3)显示增量的分布与 s 无关,所以增量具有平稳性。条件(2)和(3)充分刻画了过程前后的独立性和时间上的均匀性。强度 λ 有时也称为速率,它描绘随机事件发生的频繁程度。

例 8.3.1 顾客依泊松过程到达某商店,速率为 $\lambda=4$ 人/h。已知商店上午 9:00 开门。试求到 9:30 时仅到一位顾客,而到 11:30 时总计已到达 5 位顾客的概率。

解:令 t 的计时单位为 h,并以 9:00 为起始时刻,所求事件可表示为 $\left\{N\left(\frac{1}{2}\right)=1,N\left(\frac{5}{2}\right)=5\right\}$,其概率为

$$\begin{aligned}P\left\{N\left(\frac{1}{2}\right)=1,N\left(\frac{5}{2}\right)=5\right\}&=P\left\{N\left(\frac{1}{2}\right)=1,N\left(\frac{5}{2}\right)-N\left(\frac{1}{2}\right)=4\right\}\\&=\left\{\frac{\mathrm{e}^{-4\cdot\frac{1}{2}}\cdot 4\cdot\frac{1}{2}}{1!}\right\}\left\{\frac{\mathrm{e}^{-4\cdot 2}\cdot(4\cdot 2)^4}{4!}\right\}=0.0155\end{aligned}$$

定义 8.3.2 $\{W(t),t\geqslant 0\}$是二阶矩过程,如果它满足下述 4 个条件就称其为**维纳过程**:

(1) 具有平稳的独立增量;

(2) 对任意的 $t>s\geqslant 0$,$W(t)-W(s)$服从正态分布;

(3) 对任意的 $t\geqslant 0$,$E[W(t)]=0$;

(4) $W(0)=0$。

条件(1)显示维纳过程是齐次的独立增量过程,它也是正态过程,其分布完全由均值函数和相关函数(或协方差函数)确定。依据条件(3)$E[W(t)]=0$,进一步可以证明 $D_{\mathrm{W}}(t)=E[W^2(t)]=\sigma^2 t$(其中 σ^2 是维纳过程的参数,可以通过实验观察值加以估计)。再根据式(8.12)可以求得协方差函数(相关函数)为

$$C_{\mathrm{WW}}(s,t)=R_{\mathrm{WW}}(s,t)=\sigma^2\min(s,t)$$

8.4 马尔柯夫过程

马尔柯夫过程因俄国数学家 Markov 的研究而得名。马尔柯夫过程在近代物理、生物学、信息处理及计算方法等方面都有重要应用。在物理学中,很多确定性现象遵从以下演变原则:有时刻 t 过程所处的状态,可以决定过程在时刻 $s>t$ 所处的状

态，而无需借助于 t 以前过程所处状态的历史资料。由此引入了随机过程中的**马尔柯夫性**或**无后效性**。

过程在时刻 t 所处状态为已知的条件下，过程在时刻 $s>t$ 所处状态的条件分布与过程在时刻 t 之前所处的状态无关。通俗地说，就是在已经知道过程“现在”的条件下，其“将来”不依赖于“过去”。

8.4.1　马尔柯夫过程及其概率分布

设随机过程 $\{X(t),t\in T\}$ 的状态空间为 I。如果对时间 t 的任意 n 个数值，$t_1<t_2<\cdots<t_n$，$n\geqslant 3$，$t_i\in T$，在条件 $X(t_i)=x_i$，$x_i\in I$，$i=1,2,\cdots,n-1$ 下，$X(t_n)$ 的条件分布函数恰等于在条件 $X(t_{n-1})=x_{n-1}$ 下 $X(t_n)$ 的条件分布函数，即

$$\begin{aligned} &P\{X(t_n)\leqslant x_n \mid X(t_1)=x_1,X(t_2)=x_2,\cdots,X(t_{n-1})=x_{n-1}\} \\ &\quad = P\{X(t_n)\leqslant x_n \mid X(t_{n-1})=x_{n-1}\} \quad x_n\in R \end{aligned} \tag{8.14}$$

则称随机过程 $\{X(t),t\in T\}$ 为**马尔柯夫过程**。

时间和状态都是离散的马尔柯夫过程称为**马尔柯夫链**，简称**马氏链**，记为 $\{X_n=X(n),n=0,1,2,\cdots\}$，它可以看作在时间集 $T_1=\{0,1,2,\cdots\}$ 上对离散状态的过程相继观察的结果。约定链的状态空间为 $I=\{a_1,a_2,\cdots\}$，$a_i\in R$。在链的情形，马尔柯夫性通常用条件分布来表示，即对任意的正整数 n、r 和 $0\leqslant t_1<t_2<\cdots<t_r<m$，$t_i$、$m$、$n+m\in T_1$，有

$$\begin{aligned} &P\{X_{m+n}=a_j \mid X_{t_1}=a_{i_1},X_{t_2}=a_{i_2},\cdots,X_{t_r}=a_{i_r},X_m=a_i\} \\ &=P\{X_{m+n}=a_j \mid X_m=a_i\} \end{aligned} \tag{8.15}$$

则称条件概率

$$P_{ij}(m,m+n)=P\{X_{m+n}=a_j \mid X_m=a_i\} \tag{8.16}$$

为马尔柯夫链在时刻 m 处于状态 a_i 条件下，在时刻 $m+n$ 转移到状态 a_j 的**转移概率**。

由于马尔柯夫链在时刻 m 从任何状态 a_i 出发，到另一个时刻 $m+n$，必然转移到 $a_1,a_2,\cdots$ 诸状态中的某一个，所以

$$\sum_{j=1}^{\infty} P_{ij}(m,m+n)=1,\quad i=1,2,\cdots \tag{8.17}$$

由转移概率组成的矩阵 $\boldsymbol{P}(m,m+n)\equiv(P_{ij}(m,m+n))$ 称为马尔柯夫链的**转移概率矩阵**。由式(8.17)知，此矩阵的每一行元素和等于 1。

当转移概率 $P_{ij}(m,m+n)$ 只与 i、j 及时间间距 n 有关时，即 $P_{ij}(m,m+n)=P_{ij}(n)$ 时，称转移概率具有**平稳性**。同时也称此链是**齐次的**。

在马尔柯夫链为齐次的情形下，由式(8.16)定义的转移概率

$$P_{ij}(n)=P\{X_{m+n}=a_j \mid X_m=a_i\} \tag{8.18}$$

称为马尔柯夫链的 n **步转移概率**，$P(n)\equiv(P_{ij}(n))$ 为 n **步转移概率矩阵**。在以下的讨论中特别重要的是**一步转移概率**

$$p_{ij}=P_{ij}(1)=P\{X_{m+1}=a_j \mid X_m=a_i\}$$

或由它们组成的**一步转移概率矩阵**

$$\begin{array}{cc|ccccc} & & & & X_{m+1}\text{ 的状态} & & \\ & & a_1 & a_2 & \cdots & a_j & \cdots \\ X_m & a_1 & p_{11} & p_{12} & \cdots & p_{1j} & \cdots \\ \text{的} & a_2 & p_{21} & p_{22} & \cdots & p_{2j} & \cdots \\ \text{状} & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \text{态} & a_i & p_{i1} & p_{i2} & \cdots & p_{ij} & \cdots \\ & \vdots & & & & & \end{array} = \boldsymbol{P}(1) = \boldsymbol{P}$$

在上述矩阵的左侧和上面标上状态 $a_1,a_2,\cdots$是为了显示 p_{ij} 是由状态 a_i 经一步转移到状态 a_j 的概率。

例 8.4.1(一维随机游动) 设一醉汉 Q(或看作一随机游动的质点),在如图 8.2 所示直线的点集 $I=\{1,2,3,4,5\}$上做随机游动,并且仅仅在 1s、2s 等时刻发生游动。游动的概率规则是:如果 Q 现在位于点 $i(1<i<5)$,则下一时刻各以$\frac{1}{3}$的概率向左或向右移动一格,或以$\frac{1}{3}$的概率留在原处;如果 Q 现在位于 1(或 5)点上,则下一时刻就以概率 1 移动到 2(或 4)点上。1 和 5 这两个点称为反射壁。上面这种游动称为带有两个反射壁的随机游动。

图 8.2　例 8.4.1 用图

若以 X_n 表示时刻 n 时 Q 的位置,不同的位置就是 X_n 的不同状态,那么$\{X_n,n=0,1,2,\cdots\}$是一随机过程,状态空间就是 $I=\{1,2,3,4,5\}$,而且当 $X_n=i,i\in I$ 为已知时,X_{n+1}所处的状态的概率分布只与 $X_n=i$ 有关,而与 Q 时刻 n 以前如何到达 i 是完全无关的,所以$\{X_n,n=0,1,2,\cdots\}$是一马尔柯夫链,而且还是齐次的。它的一步转移概率和一步转移概率矩阵分别为

$$p_{ij}=P\{X_{n+1}=j\mid X_n=i\}=\begin{cases}\frac{1}{3}, & j=i-1,i,i+1;1<i<5\\ 1, & i=1,j=2;i=5,j=4\\ 0, & |j-i|\geqslant 2\end{cases}$$

和

$$\boldsymbol{P}=\begin{bmatrix}0 & 1 & 0 & 0 & 0\\ \frac{1}{3} & \frac{1}{3} & \frac{1}{3} & 0 & 0\\ 0 & \frac{1}{3} & \frac{1}{3} & \frac{1}{3} & 0\\ 0 & 0 & \frac{1}{3} & \frac{1}{3} & \frac{1}{3}\\ 0 & 0 & 0 & 1 & 0\end{bmatrix}$$

如果把 1 这一点改为吸收壁,即是说 Q 一旦到达 1 这一点,则就永远留在 1 上。此时,相应链的转移概率矩阵只需把 $\boldsymbol{P}$ 中的第 1 行改为(1,0,0,0,0)。总之,改变游动的概率规则,就可得到不同方式的随机游动和相应的马尔柯夫链。

例 8.4.2(**排队模型**)　设服务系统由一个服务员和只可以容纳两个人的等候室组成,见图 8.3。服务规则是:先到先服务,后来者需在等候室依次排队。假定一个需要服务的顾客到达系统时发现系统内已有 3 个顾客(一个正在接受服务,两个在等候室排队),则该顾客即离去。设时间间隔 Δt 内将有一个顾客进入系统的概率为 q,有一原来被服务的顾客离开系统(即服务完毕)的概率为 p。又设当 Δt 充分小时,在这时间间隔内多于一个顾客进入或离开系统实际上是不可能的。再设有无顾客来到与服务是否完毕是相互独立的。现在用马尔柯夫链描述这个服务系统。

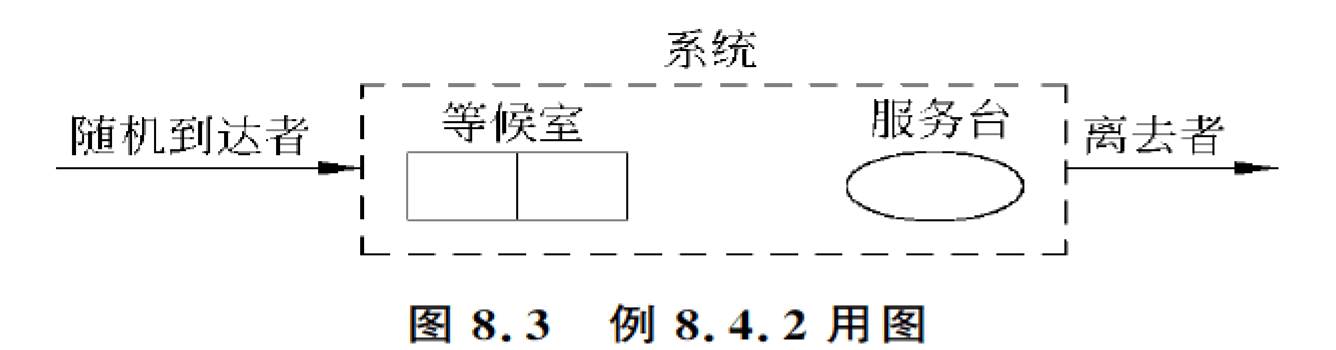


图 8.3　例 8.4.2 用图

设 $X_n = X(n\Delta t)$ 表示时刻 $n\Delta t$ 时系统内的顾客数,即系统的状态。$\{X_n, n=0,1,2,\cdots\}$ 是一随机过程,状态空间就是 $I=\{0,1,2,3\}$,而且仿照例 8.4.1 的分析,可知它是一个齐次马尔柯夫链。下面来计算此马尔柯夫链的一步转移概率。

p_{00}——在系统内没有顾客的条件下,经 Δt 后仍没有顾客的概率(此处是条件概率,以下同),$p_{00}=1-q$。

p_{01}——在系统内没有顾客的条件下,经 Δt 后有一顾客进入系统的概率,$p_{01}=q$。

p_{10}——系统内恰有一顾客正在接受服务的条件下,经 Δt 后系统内无人的概率。它等于在 Δt 间隔内顾客因服务完毕而离去,且无人进入系统的概率,$p_{10}=p(1-q)$。

p_{11}——系统内恰有一顾客的条件下,在 Δt 间隔内,他因服务完毕而离去,而另一顾客进入系统;或者正在接受服务的顾客将继续要求服务,且无人进入系统的概率,$p_{11}=pq+(1-p)(1-q)$。

p_{12}——正在接受服务的顾客继续要求服务,且另一顾客进入系统的概率,$p_{12}=q(1-p)$。

p_{13}——正在接受服务的顾客继续要求服务,且在 Δt 间隔内有两个顾客进入系统的概率。由假设,后者实际上是不可能发生的,$p_{13}=0$。

类似地,有 $p_{21}=p_{32}=p(1-q)$,$p_{22}=pq+(1-p)(1-q)$,$p_{23}=q(1-p)$, $p_{ij}=0(|i-j|\geqslant 2)$。

p_{33}——或者一人将离去且另一人将进入系统,或者无人离开系统的概率,$p_{33}=pq+(1-p)$。

于是该马尔柯夫链的一步转移概率矩阵为

$$\boldsymbol{P}=\begin{bmatrix} 1-q & q & 0 & 0 \\ p(1-q) & pq+(1-p)(1-q) & q(1-p) & 0 \\ 0 & p(1-q) & pq+(1-p)(1-q) & q(1-p) \\ 0 & 0 & p(1-q) & pq+(1-p) \end{bmatrix}$$

在实际中,一步转移概率通常可以通过统计试验确定,下面看一实例。

例 8.4.3　某台计算机经常出故障,研究者每隔 15min 观察一次计算机的运行

状态，收集了 24h 的数据（共作 97 次观察）。用 1 表示正常状态，用 0 表示不正常状态，所得的数据序列如下：

11100100 11111110 01111011 11110011 11111110 001101101
11101101 10101111 01110111 10111111 00110111 11100111

设 X_n 为第 $n(n=1,2,\cdots,97)$ 个时段的计算机状态，可以认为它是一个齐次马尔柯夫链，状态空间 $I=\{0,1\}$。96 次状态转移的情况是：

$0\to0$，8 次； $0\to1$，18 次； $1\to0$，18 次； $1\to1$，52 次。

因此，一步转移概率可用频率近似地表示为

$$p_{00}=P\{X_{n+1}=0\mid X_n=0\}\approx\frac{8}{8+18}=\frac{8}{26}$$

$$p_{01}=P\{X_{n+1}=1\mid X_n=0\}\approx\frac{18}{8+18}=\frac{18}{26}$$

$$p_{10}=P\{X_{n+1}=0\mid X_n=1\}\approx\frac{18}{18+52}=\frac{18}{70}$$

$$p_{11}=P\{X_{n+1}=1\mid X_n=1\}\approx\frac{52}{18+52}=\frac{52}{70}$$

下面就来研究齐次马尔柯夫链的有限维分布。先看马尔柯夫链在任一时刻 $n\in T_1$ 的一维分布：

$$p_j(n)=P\{X_n=a_j\},\quad a_j\in I,j=1,2,\cdots \tag{8.19}$$

显然，应该有 $\sum_{j=1}^{\infty}p_j(n)=1$。由全概率公式，又有

$$P\{X_n=a_j\}=\sum_{i=1}^{\infty}P\{X_n=a_j\mid X_0=a_i\}P\{X_0=a_i\}$$

或

$$p_j(n)=\sum_{i=1}^{\infty}p_i(0)P_{ij}(n),\quad j=1,2,\cdots \tag{8.20}$$

一维分布式(8.19)也可用行向量表示成

$$p(n)=(p_1(n),p_2(n),\cdots,p_j(n),\cdots) \tag{8.21}$$

这样，可以利用矩阵乘法（I 是可列无限集时，仍可以用有限阶矩阵乘法的规则确定矩阵之积的元素），式(8.20)可以写成

$$p(n)=p(0)P(n) \tag{8.22}$$

此式表明，马尔柯夫链在任一时刻 $n\in T_1$ 的一维分布由初始分布 $p(0)$和 n 步转移概率矩阵所确定。

又对于任意 n 个时刻 $t_1<t_2<\cdots<t_n(t_i\in T_1)$以及状态 $a_{i_1},a_{i_2},\cdots(a_{i_n}\in I)$，马尔柯夫链的 n 维分布为

$$\begin{aligned}
&P\{X_{t_1}=a_{i_1},X_{t_2}=a_{i_2},\cdots,X_{t_n}=a_{i_n}\}\\
=&P\{X_{t_1}=a_{i_1}\}\cdot P\{X_{t_2}=a_{i_2}\mid X_{t_1}=a_{i_1}\}\\
&\cdot P\{X_{t_3}=a_{i_3}\mid X_{t_1}=a_{i_1},X_{t_2}=a_{i_2}\}\cdots\\
&\cdot P\{X_{t_n}=a_{i_n}\mid X_{t_1}=a_{i_1},X_{t_2}=a_{i_2},\cdots,X_{t_{n-1}}=a_{i_{n-1}}\}
\end{aligned}$$

$$=P\{X_{t_1}=a_{i_1}\}\cdot P\{X_{t_2}=a_{i_2}\mid X_{t_1}=a_{i_1}\}$$
$$\cdot P\{X_{t_3}=a_{i_3}\mid X_{t_2}=a_{i_2}\}\cdots\cdot P\{X_{t_n}=a_{i_n}\mid X_{t_{n-1}}=a_{i_{n-1}}\}$$
$$=p_{i_1}(t_1)P_{i_1i_2}(t_2-t_1)P_{i_2i_3}(t_3-t_2)\cdots P_{i_{n-1}i_n}(t_n-t_{n-1}) \tag{8.23}$$

式(8.23)的第一个等式利用了乘法定理,第二个等式利用了马尔柯夫性,第三个等式利用了齐次性。由此,结合式(8.20)和式(8.23)可知,马尔柯夫链的有限维分布同样完全由初始分布和转移概率确定。

总之,转移概率决定了马尔柯夫链的统计规律。因此,确定马尔柯夫链的任意 n 步转移概率就成为马尔柯夫链研究中的重要问题之一。

例 8.4.4(续例 8.4.3)　若计算机在前一段时段(15min)的状态为 0,问从本时段起此计算机能连续正常工作 1h(4 个时段)的概率为多少?

解: 由题意,前一段时段的状态为 0 就是初始分布 $p_0(0)=P\{X_0=0\}=1$。于是由式(8.23),计算机能连续正常工作 4 个时段的概率为

$$P\{X_0=0,X_1=1,X_2=1,X_3=1,X_4=1\}$$
$$=p_0(0)P_{01}(1)P_{11}(1)P_{11}(1)P_{11}(1)$$
$$=1\cdot\frac{18}{26}\cdot\frac{52}{70}\cdot\frac{52}{70}\cdot\frac{52}{70}=0.284$$

式中数据见例 8.4.3。

8.4.2　多步转移概率的确定

为了确定齐次马尔柯夫链的 n 步转移概率 $P_{ij}(n)$,首先介绍 $P_{ij}(n)$所满足的基本方程。设$\{X(n),n\in T_1\}$是一齐次马尔柯夫链,则对于任意的 $u,v\in T_1$,有

$$P_{ij}(u+v)=\sum_{k=1}^{\infty}P_{ik}(u)P_{kj}(v),\quad i,j=1,2,\cdots \tag{8.24}$$

方程(8.24)是著名的切普曼-柯莫洛夫(Chapman-Kolmogorov)方程,简称 C-K 方程。

C-K 方程基于下述事实,即"从时刻 s 所处的状态 a_i,即 $X(s)=a_i$ 出发,经时段 $u+v$ 转移到状态 a_j,即 $X(s+u+v)=a_j$"这一事件可分解成"从 $X(s)=a_i$ 出发,先经时段 u 转移到中间状态 $a_k(k=1,2,\cdots)$,再从 a_k 经时段 v 转移到状态 a_j"这样一些事件的和事件,见图 8.4。

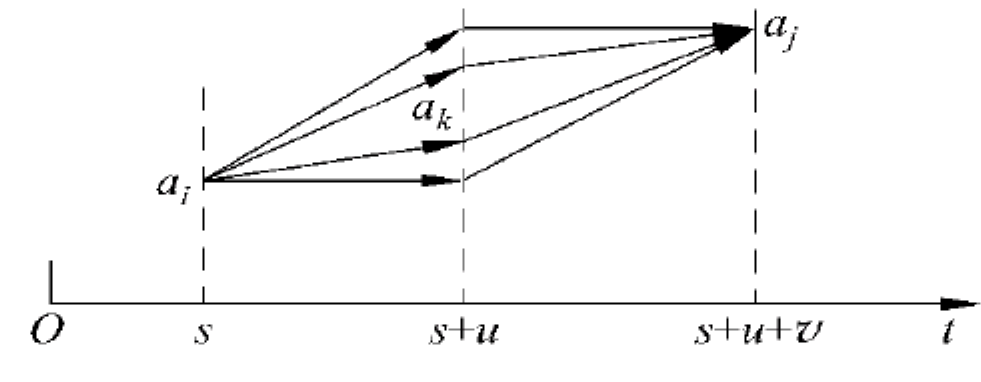


图 8.4　C-K 方程的原理

方程(8.24)的证明如下:先固定 $a_k\in I$ 和 $s\in T_1$,由条件概率、乘法定理、马尔柯夫性和齐次性可得

$$P\{X(s+u+v)=a_j,X(s+u)=a_k\mid X(s)=a_i\}$$
$$=P\{X(s+u)=a_k\mid X(s)=a_i\}$$
$$\cdot P\{X(s+u+v)=a_j\mid X(s+u)=a_k,X(s)=a_i\}$$
$$=P_{ik}(u)\cdot P_{kj}(v) \tag{8.25}$$

又由于事件组“$X(s+u)=a_k$”，$k=1,2,\cdots$构成一划分，故有

$$P_{ij}(u+v) = P\{X(s+u+v) = a_j \mid X(s) = a_i\}$$
$$= \sum_{k=1}^{\infty} P\{X(s+u+v) = a_j, X(s+u) = a_k \mid X(s) = a_i\}$$

将式(8.25)代入上式，即得所要证明的C-K方程。

C-K方程也可写成矩阵形式：

$$P(u+v) = P(u)P(v) \tag{8.26}$$

利用C-K方程容易确定n步转移概率。事实上，在式(8.26)中，令$u=1$，$v=n-1$，得递推关系：

$$P(n) = P(1)P(n-1) = PP(n-1)$$

从而可得

$$P(n) = P^n \tag{8.27}$$

就是说，对齐次马尔柯夫链而言，n步转移概率矩阵是一步转移概率矩阵的n次方。进而可知，链的有限分布可由初始分布与一步转移概率完全确定。

例 8.4.5 设$\{X_n, n\geqslant 0\}$是具有3个状态0、1、2的齐次马尔柯夫链，一步转移概率矩阵为

$$\boldsymbol{P} = \begin{bmatrix} \frac{3}{4} & \frac{1}{4} & 0 \\ \frac{1}{4} & \frac{1}{2} & \frac{1}{4} \\ 0 & \frac{3}{4} & \frac{1}{4} \end{bmatrix}$$

初始分布$p_i(0)=P\{X_0=i\}$，$i=0,1,2$。试求：(1)$P(X_0=0, X_2=1)$；(2)$P(X_2=1)$。

解：先求出二步转移概率矩阵为

$$P(2) = \boldsymbol{P}^2 = \begin{bmatrix} \frac{5}{8} & \frac{5}{16} & \frac{1}{16} \\ \frac{5}{16} & \frac{1}{2} & \frac{3}{16} \\ \frac{3}{16} & \frac{9}{16} & \frac{1}{4} \end{bmatrix}$$

于是(1)

$$P\{X_0 = 0, X_2 = 1\} = P\{X_0 = 0\}P\{X_2 = 1 \mid X_0 = 0\} = p_0(0)p_{01}(2)$$
$$= \frac{1}{3} \cdot \frac{5}{16} = \frac{5}{48}.$$

(2) 由式(8.20)，

$$p_1(2) = P\{X_2 = 1\} = p_0(0)p_{01}(2) + p_1(0)p_{11}(2) + p_2(0)p_{21}(2)$$
$$= \frac{1}{3}\left[\frac{5}{16} + \frac{1}{2} + \frac{9}{16}\right] = \frac{11}{24}$$

8.4.3 马尔柯夫链的平稳分布

设齐次马尔柯夫链$\{X_n, n\geqslant 1\}$的状态空间为I，$\boldsymbol{P}$是它的一步转移概率矩阵；如

果对于所有 $a_i,a_j\in I$,转移概率 $P_{ij}(n)$存在极限

$$\lim_{n\to\infty}P_{ij}(n)=\pi_j\quad(\text{不依赖于 } i)$$

则称此链具有**遍历性**。又若 $\sum_j \pi_j=1$,则同时称 $\Pi=(\pi_1,\pi_2,\cdots)$为链$\{X_n,n\geqslant 1\}$的**极限分布**。

设 $\Gamma=(\upsilon_1,\upsilon_2,\cdots)$是 I 上的一概率分布,如果有 $\Gamma=\Gamma\cdot P$,即有 $\upsilon_j=\sum_{i:a_i\in I}\upsilon_i p_{ij}$,则称 $\Gamma=(\upsilon_1,\upsilon_2,\cdots)$为链$\{X_n,n\geqslant 1\}$的**平稳分布**。

齐次马尔柯夫链在什么条件下才具有遍历性?如何求出它的极限分布和平稳分布?下面仅就只有有限个状态的链,即**有限状态链**的遍历性给出一个充分条件。

定理 8.4.1　设齐次马尔柯夫链$\{X_n,n\geqslant 1\}$的状态空间为 $I=\{a_1,a_2,\cdots,a_N\}$,$\boldsymbol{P}$ 是它的一步转移概率矩阵,如果存在正整数 m,使得对任意 $a_i,a_j\in I$,都有

$$P_{ij}(m)>0,\quad i,j=1,2,\cdots,N \tag{8.28}$$

则此链具有遍历性;且有极限分布 $\Pi=(\pi_1,\pi_2,\cdots,\pi_N)$,它是方程组

$$\Pi=\Pi\cdot\boldsymbol{P}\quad\text{或即}\quad \pi_j=\sum_{i=1}^{N}\pi_i p_{ij},\quad j=1,2,\cdots,N \tag{8.29}$$

的满足条件

$$\pi_j>0,\quad \sum_{j=1}^{N}\pi_j=1 \tag{8.30}$$

的唯一解。

依照定理,为证明有限链是遍历的,只需找一正整数 m,使 m 步转移概率矩阵 $\boldsymbol{P}^m$ 无零元。而求极限分布 $\Pi=(\pi_1,\pi_2,\cdots,\pi_N)$的问题,化为求解方程组(8.29)的问题。在定理的条件下,齐次马尔柯夫链$\{X_n,n\geqslant 1\}$的极限分布就是它的**平稳分布**。

容易看出,如果 $\Pi=(\pi_1,\pi_2,\cdots,\pi_N)$为链的初始分布,即 $P\{X_1=a_j\}=\pi_j$,则有

$$P\{X_2=a_j\}=\sum_{i=1}^{\infty}P\{X_2=j\mid X_1=i\}P\{X_1=i\}=\sum_{i=1}^{\infty}\pi_i P_{ij}=\pi_j$$

并由归纳法可得

$$P\{X_n=a_j\}=\sum_{i=1}^{\infty}P\{X_n=j\mid X_{n-1}=i\}P\{X_{n-1}=i\}=\sum_{i=1}^{\infty}\pi_i P_{ij}=\pi_j$$

于是对所有 n,X_n 有相同分布,即 $p(n)$永远和 $\Pi=(\pi_1,\pi_2,\cdots,\pi_N)$一致。

例 8.4.6　试说明例 8.4.2 的排队模型中的链是遍历的,并求其极限分布。

解:由例 8.4.2 中的一步转移概率矩阵 $\boldsymbol{P}$,可算得 $P(3)=\boldsymbol{P}^3$ 无零元。依据定理 8.4.1,该链是遍历的。而极限分布 $\Pi=(\pi_0,\pi_1,\pi_2,\pi_3)$满足下列方程组

$$\begin{cases}\pi_0=(1-q)\pi_0+p(1-q)\pi_1\\ \pi_1=q\pi_0+[pq+(1-p)(1-q)]\pi_1+[p(1-q)]\pi_2\\ \pi_2=q(1-p)\pi_1+[pq+(1-p)(1-q)]\pi_2+[p(1-q)]\pi_3\\ \pi_3=q(1-p)\pi_2+[pq+(1-p)]\pi_3\\ \pi_0+\pi_1+\pi_2+\pi_3=1\end{cases}$$

解之,得唯一解

$$\pi_0 = \frac{p^3(1-q)^3}{C}$$

$$\pi_1 = \frac{p^2q(1-q)^2}{C}$$

$$\pi_2 = \frac{pq^2(1-q)(1-p)}{C}$$

$$\pi_3 = \frac{q^3(1-p)^2}{C}$$

其中 $$C = p^3(1-q)^3 + p^2q(1-q)^2 + pq^2(1-q)(1-p) + q^3(1-p)^2$$

有限状态马尔柯夫链具有遍历性的判别是随后讨论马尔柯夫密码的关键，所以下面继续给出另外的充分必要条件。

定义 8.4.1 设马尔柯夫链$\{X_n, n \geqslant 1\}$的状态空间为 $I=\{a_1, a_2, \cdots\}$，对 $a_i, a_j \in I$，若存在自然数 m，使得 $P_{ij}(m)>0$，则称自状态 a_i 出发可达状态 a_j，记为 $a_i \to a_j$。如果 $a_i \to a_j$ 且 $a_j \to a_i$，则称 a_i 和 a_j 是**相通的**，记为 $a_i \leftrightarrow a_j$。如果马尔柯夫链$\{X_n, n \geqslant 1\}$的任意两个状态都相通，则称为**不可约链**。

如果两个状态 a_i 和 a_j 不是相通的，那么就有对所有的 $m \geqslant 1$，$P_{ij}(m)=0$ 或者对所有的 $m \geqslant 1$，$P_{ji}(m)=0$，或者两者都成立。三者情况必居其一。相通性是一种数学上的等价关系，也就是说它满足自反性、对称性和传递性。两个状态如果是相通的，就称它们是处在同一类中。马尔柯夫链的所有状态由相通这一等价关系分割成不同的等价类。不可约链就是在相通这一等价关系下所有状态都属于同一类的马尔柯夫链，换言之，不可约过程的各个状态都是相通的。

定义 8.4.2 设马尔柯夫链$\{X_n, n \geqslant 1\}$的状态空间为 $I=\{a_1, a_2, \cdots\}$，对状态 a_i，如果集合$\{m: m \geqslant 1, P_{ii}(m)>0\} \neq \varnothing$，称该数集的最大公约数 $d(i)$为状态 a_i 的周期。如果 $d(i)>1$，称状态 a_i 为**周期的**；如果 $d(i)=1$，称状态 a_i 为**非周期的**。

由定义立即可知，如果 m 不能被周期 $d(i)$整除，则必有 $P_{ii}(m)=0$。

例 8.4.7 马尔柯夫链有状态 0、1、2、3，转移概率矩阵为

$$\boldsymbol{P} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ \frac{1}{2} & 0 & \frac{1}{2} & 0 \end{bmatrix}$$

试求状态 0 的周期。

解：不难计算出 $P_{00}=0$，$P_{00}(2)=P_{00}(3)=P_{00}(5)=P_{00}(2m+1)=0$，$P_{00}(4)=\frac{1}{2}$，$P_{00}(6)=\frac{1}{4}$，$P_{00}(8)=\frac{3}{8}$，而$\{4,6,8,\cdots\}$的最大公约数为 2，所以 $d(0)=2$。

那么上例中其他状态的周期是多少呢？由于周期性是一种整个等价类所具有的性质，所以如果过程是不可约的，则每个状态都有相同的周期。这就是下面的引理。

引理 8.4.1 设马尔柯夫链$\{X_n, n \geqslant 1\}$的状态空间为 $I=\{a_1, a_2, \cdots\}$，如果 $a_i \leftrightarrow a_j$，则 $d(i)=d(j)$。

证明：由 $a_i \leftrightarrow a_j$ 知，存在 m,n 使 $P_{ij}(m)>0$ 和 $P_{ji}(n)>0$。于是有

$$P_{jj}(n+m) \geqslant P_{ji}(n)P_{ij}(m) > 0$$

假如有 s 使 $P_{ii}(s)>0$，则也有

$$P_{jj}(n+s+m) \geqslant P_{ji}(n)P_{ii}(s)P_{ij}(m) > 0$$

因为不等式最左边所表示的从状态 a_j 出发经过 $n+s+m$ 步转移后又回到 a_j 的概率，它当然要大于一个加了更多限制的子事件的概率。这个子事件是从 a_j 出发经过 n 步转移到 a_i，再经过 s 步返回到 a_i，又再从 a_i 出发经过 m 步到达 a_j。它的效果也是转移 $n+s+m$ 步回到 a_j。由 $d(j)$ 的定义，它将同时整除 $n+m$ 及 $n+s+m$，所以 $d(j)$ 必整除 s，而 $d(i)$ 是所有使 $P_{ii}(s)>0$ 的 s 的最大公约数，所以 $d(j)$ 整除 $d(i)$。同样可证 $d(i)$ 整除 $d(j)$，所以有 $d(i)=d(j)$。

引理 8.4.2　设马尔柯夫链 $\{X_n, n\geqslant 1\}$ 的状态空间为 $I=\{a_1,a_2,\cdots\}$，如果状态 a_i 有周期 $d(i)$，则存在整数 M，使得对所有的 $n>M$ 恒有 $P_{ii}(nd(i))>0$。

证明：数论中有一个事实：如果正整数 $n_1,n_2,\cdots,n_k$ 的最大公约数为 d，则存在正整数 M 使得对所有的 $n>M$，能找到非负整数 c_t 使得

$$nd = \sum_{t=1}^{k} c_t n_t \tag{8.31}$$

对于状态 a_i，令 $n_1,n_2,\cdots,n_k$ 为使 $P_{ii}(n_t)>0, t=1,2,\cdots,k$ 成立的正整数。由周期的定义，$d(i)$ 是它们的最大公约数。而由式(8.31)知，对 $n>M$，有

$$P_{ii}(nd(i)) = P_{ii}\left(\sum_{t=1}^{k} c_t n_t\right) \geqslant \prod_{t=1}^{k} (P_{ii}(n_t))^{c_t} > 0$$

由引理 8.4.2 及事实 $P_{ii}(m+nd(i)) \geqslant P_{ji}(m)P_{ii}(nd(i))$ 立即有下面的推论。

推论 8.4.1　如果 $P_{ji}(m)>0$，则存在正整数 M，使得对所有的 $n>M$ 恒有 $P_{ji}(m+nd(i))>0$。

讨论这些性质的意义何在呢？它可以帮助我们研究当 n 很大时 $P_{ji}(n)$ 的极限是否存在？存在的条件是什么？如果存在又如何简便地求出它们？周期性是一种等价类中全体状态共有的性质。当周期为 1 时，马尔柯夫链称为是**非周期的**。对于非周期不可约的马尔柯夫链有以下定理。

定理 8.4.2　设马尔柯夫链 $\{X_n, n\geqslant 1\}$ 的状态空间为 $I=\{a_1,a_2,\cdots,a_N\}$，且该链是不可约和非周期的，$\boldsymbol{P}$ 是它的一步转移概率矩阵，则必存在 M，使得当 $n\geqslant M$ 时，n 步转移概率矩阵的所有元素都非零。

证明：由于马尔柯夫链 $\{X_n, n\geqslant 1\}$ 是不可约的，过程的任两个状态 a_i 和 a_j 都是相通的，于是存在 m(与 a_i 和 a_j 有关)使 $P_{ij}(m)>0$。由推论 8.4.1 及链非周期得知：存在 M，使得当 $n\geqslant M$ 时有 $P_{ij}(m+n\cdot 1)>0$。因状态空间是有限的，对全部的状态对 (a_i,a_j) 求出 $M(i,j)$。并取 $M=\max\limits_{(i,j)}(m(i,j)+M(i,j))$，则显然对所有状态 a_i 和 a_j，当 $n\geqslant M$ 时有 $P_{ij}(n)>0$。

8.5　马尔柯夫密码

如果分组密码 E 是基于一个简单函数 F 迭代若干次而形成，如图 8.5 所示，就称其为迭代密码。每次迭代称为一轮。相应的函数 F 称为轮函数。每一轮输出都

是前一轮输出的函数，即 $Y(i)=F(Y(i-1),K(i))$，其中 $K(i)$ 是第 i 轮子密钥，由秘密密钥 K 通过密钥扩展算法产生。DES 为 16 轮，AES-128 为 10 轮。本章假定轮函数 $F:\{0,1\}^n\times\{0,1\}^k\to\{0,1\}^n$，对每个轮子密钥 $K(i)$，$F(\circ,K(i))$ 是 $\{0,1\}^n$ 上的置换。

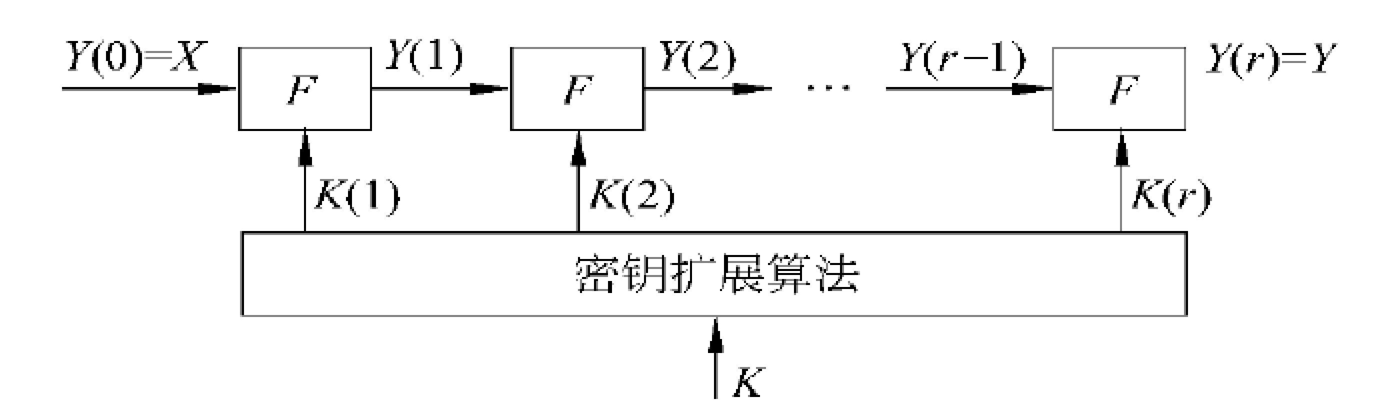


图 8.5　以 F 为轮函数的 r 轮迭代密码

迭代密码与分组密码的基本设计原则相符。简单的轮函数可方便地实现，并且适当选择的轮函数经过若干次迭代后可以提供必要的混乱和扩散，所以目前流行的分组密码均是迭代型密码。

对于任意两个 n 比特串 X 和 X^*，它们的**差分**定义为

$$\Delta X = X\otimes(X^*)^{-1}$$

其中，$\otimes$ 表示 $\{0,1\}^n$ 上的一个群运算；$(X^*)^{-1}$ 表示元素 X^* 在群 $(\{0,1\}^n,\otimes)$ 中的逆元。用 e 表示群 $(\{0,1\}^n,\otimes)$ 的幺元。

定义 8.5.1　对轮函数 $F:\{0,1\}^n\times\{0,1\}^k\to\{0,1\}^n$，$F(X,Z)=Y$，如果存在用来定义差分的群运算 $\otimes$，使得对任意 $\alpha,\beta\in\{0,1\}^n\backslash\{e\}$，当子密钥 Z 是均匀随机时，

$$P\{\Delta Y=\beta\mid \Delta X=\alpha, X=\gamma\}$$

和 X 的取值 γ 无关；则以 F 为轮函数的迭代密码称为**马尔柯夫密码**。

用图 8.5 所示的 r 轮迭代密码加密一对不同的明文 X 和 X^*，它们的加密过程和差分序列如图 8.6 所示。

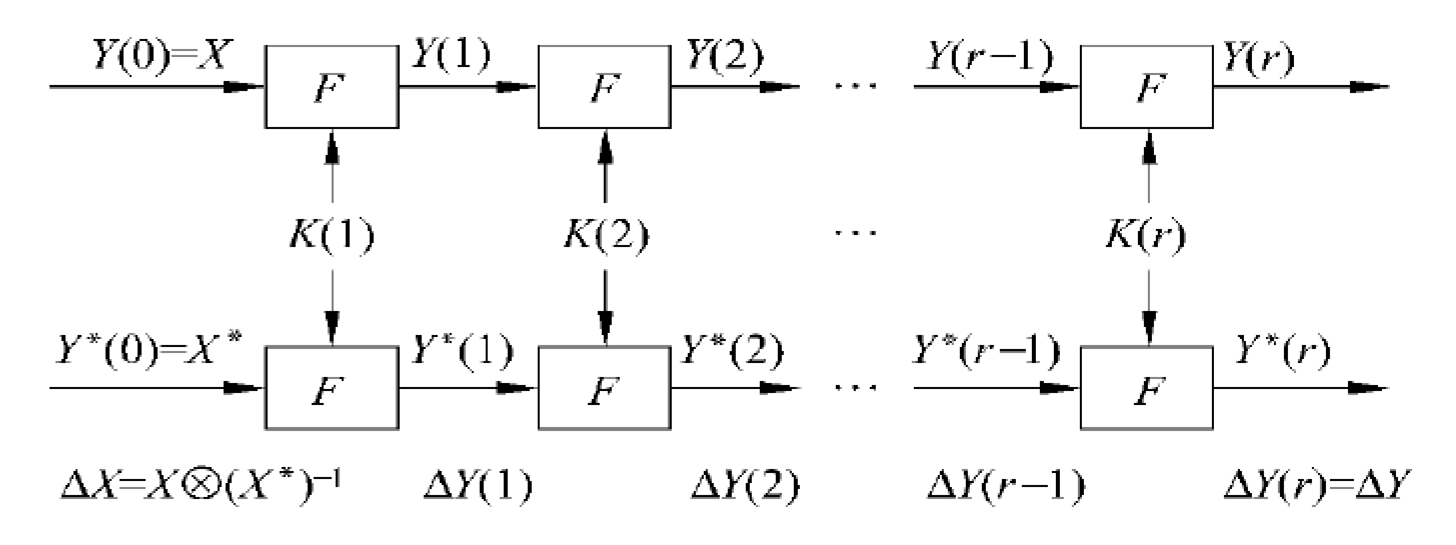


图 8.6　r 轮迭代密码加密一对明文的差分序列

由加密对可得

差分序列　　　$\Delta Y(0)$，　$\Delta Y(1)$，　…，　$\Delta Y(r)$

其中 $Y(0)=X$ 和 $Y^*(0)=X^*$ 表示明文对，使得 $\Delta Y(0)=\Delta X$，并且 $Y(i)$ 和 $Y^*(i)$ $(0<i<r)$ 是第 i 轮的输出，它们也是第 $i+1$ 轮的输入。第 i 轮的子密钥记为 $K(i)$，F 是轮函数，使得

$$Y(i) = F(Y(i-1), K(i))$$

在下面的讨论中,总假设 $X \neq X^*$,因为当 $X = X^*$ 时,所有 $\Delta Y(i)$ 都等于群$(\{0,1\}^n, \otimes)$的幺元 e,而这种情形对于差分密码分析没有意义,从而 $\Delta Y(i) \in \{0,1\}^n \setminus \{e\}$。也假设用于迭代密码每一轮的子密钥是统计独立且均匀分布的,实际中,这一假设将作为密钥扩展算法设计的一个目标。

下面的定理解释为什么使用"马尔柯夫密码"这一术语。

定理 8.5.1　若以 F 为轮函数的 r 轮迭代密码是马尔柯夫密码,且 r 轮子密钥是独立随机均匀的,则差分序列 $\Delta X = \Delta Y(0), \Delta Y(1), \cdots, \Delta Y(r)$ 是一条齐次马尔柯夫链;若 ΔX 在群中的非幺元素上是均匀分布的,则这条马尔柯夫链是平稳的,且均匀分布是它的平稳分布。

证明:为证明差分序列 $\Delta X = \Delta Y(0), \Delta Y(1), \cdots, \Delta Y(r)$ 是马尔柯夫链,只需证明对于第二轮满足:

$$P\{\Delta Y(2) = \beta_2 \mid \Delta Y(1) = \beta_1, \Delta X = \alpha\} = P\{\Delta Y(2) = \beta_2 \mid \Delta Y(1) = \beta_1\}$$

为证明这一点,注意到

$$\begin{aligned}
P\{\Delta Y(2) = \beta_2 \mid \Delta Y(1) = \beta_1, \Delta X = \alpha\}
&= \sum_{\gamma} P\{Y(1) = \gamma, \Delta Y(2) = \beta_2 \mid \Delta Y(1) = \beta_1, \Delta X = \alpha\} \\
&= \sum_{\gamma} P\{Y(1) = \gamma \mid \Delta Y(1) = \beta_1, \Delta X = \alpha\} \\
&\quad \cdot P\{\Delta Y(2) = \beta_2 \mid \Delta Y(1) = \beta_1, Y(1) = \gamma, \Delta X = \alpha\} \\
&= \sum_{\gamma} P\{Y(1) = \gamma \mid \Delta Y(1) = \beta_1, \Delta X = \alpha\} \\
&\quad \cdot P\{\Delta Y(2) = \beta_2 \mid \Delta Y(1) = \beta_1, Y(1) = \gamma\} \\
&= \sum_{\gamma} P\{Y(1) = \gamma \mid \Delta Y(1) = \beta_1, \Delta X = \alpha\} \\
&\quad \cdot P\{\Delta Y(2) = \beta_2 \mid \Delta Y(1) = \beta_1\} \\
&= P\{\Delta Y(2) = \beta_2 \mid \Delta Y(1) = \beta_1\}
\end{aligned}$$

其中第三个等式成立是因为 $Y(1)$ 和 $\Delta Y(1)$ 可以确定 $Y(1)$ 和 $Y^*(1)$,因此,当 $Y(1)$ 和 $\Delta Y(1)$ 给定,$\Delta Y(2)$ 不再依赖 ΔX。第三个等式成立是因为该迭代密码是马尔柯夫密码。又由于每轮使用相同的轮函数,所以差分序列 $\Delta X = \Delta Y(0), \Delta Y(1), \cdots, \Delta Y(r)$ 是齐次马尔柯夫链。

对于任意的密钥 $Z = z$,轮函数 $F(\circ, z)$ 是 $\{0,1\}^n$ 上的双射。令 $\aleph = \{(X, X^*): X, X^* \in \{0,1\}^n, X \neq X^*\}$,则 $F^2: (X, X^*) \to (F(X, z), F(X^*, z))$ 是 $\aleph$ 上的双射。如果 X 和 $\Delta X(\neq e)$ 是独立且均匀分布的,则 (X, X^*) 在 $\aleph$ 上是均匀分布的,进而 $(Y, Y^*) = (F(X, z), F(X^*, z))$ 在 $\aleph$ 上也是均匀分布的,从而 $\Delta Y(\neq e)$ 也是均匀分布的。因此,均匀分布是马尔柯夫链 $\Delta X = \Delta Y(0), \Delta Y(1), \cdots, \Delta Y(r)$ 的平稳分布。

例 8.5.1　在差分定义为 $\Delta X = X \oplus X^*$ 的情形下,分组密码 DES、AES、SMS4 等都是马尔柯夫密码,其中 $\oplus$ 表示逐比特异或。它们的证明源于下面的两个结果。

定理 8.5.2 如果迭代密码的轮函数具有形式：

$$f(X,Z)=g(X\otimes Z_A,Z_B)$$

其中$\otimes$表示$\{0,1\}^n$上的一个群运算，函数$g(\circ,Z_B)$对Z_B的每种选择均是可逆的，则此迭代密码在差分定义$\Delta X=X\otimes(X^*)^{-1}$下是一马尔柯夫密码。

证明：设$S=X\otimes Z_A$，$Y=g(S,Z_B)$，$S^*=X^*\otimes Z_A$和$Y^*=g(S^*,Z_B)$，从而有

$$\Delta S=S\otimes(S^*)^{-1}=(X\otimes Z_A)\otimes(Z_A^{-1}\otimes X^{-1})=\Delta X$$

$$\Delta Y=g(S,Z_B)\otimes(g(S^*,Z_B))^{-1}=g(S,Z_B)\otimes(g((\Delta S)^{-1}\otimes S,Z_B))^{-1}$$

因此，当给定ΔS和S时，ΔY不再依赖于X，从而有

$$\begin{aligned}&P\{\Delta Y=\beta\mid\Delta X=\alpha,X=\gamma\}\\&=P\{\Delta Y=\beta\mid\Delta S=\alpha,X=\gamma\}\\&=\sum_{\lambda}P\{\Delta Y=\beta,S=\lambda\mid\Delta S=\alpha,X=\gamma\}\\&=\sum_{\lambda}P\{\Delta Y=\beta\mid\Delta S=\alpha,X=\gamma,S=\lambda\}\cdot P\{S=\lambda\mid\Delta S=\alpha,X=\gamma\}\\&=\sum_{\lambda}P\{\Delta Y=\beta\mid\Delta S=\alpha,S=\lambda\}\cdot P\{Z_A=\gamma^{-1}\otimes\lambda\}\\&=2^{-n}\sum_{\lambda}P\{\Delta Y=\beta\mid\Delta S=\alpha,S=\lambda\}\end{aligned}$$

它与X的取值γ无关，这里证明用到子密钥$Z=(Z_A,Z_B)$是均匀随机和以下等式：

$$P\{S=\lambda\mid\Delta S=\alpha,X=\gamma\}=P\{S=\lambda\mid X=\gamma\}=P\{Z_A=\gamma^{-1}\otimes\lambda\}$$

类似地可以证明以下定理。

定理 8.5.3 如果迭代密码的轮函数具有形式：

$$f(X,Z)=g(X\otimes Z)$$

其中$\otimes$表示$\{0,1\}^n$上的一群运算，函数$g(\circ)$是$\{0,1\}^n$上的双射，则此迭代密码在差分定义$\Delta X=X\otimes(X^*)^{-1}$下是一马尔柯夫密码。

对于任何马尔柯夫密码，设$\boldsymbol{P}$表示齐次马尔柯夫链$\Delta X=\Delta Y(0),\Delta Y(1),\cdots,\Delta Y(r)$的转移概率矩阵，$\boldsymbol{P}$的第$(i,j)$个元为$P(\Delta Y(1)=\alpha_j\mid\Delta X=\alpha_i)$，其中$\alpha_1,\alpha_2,\cdots,\alpha_{2^n-1}$是$\Delta X$的$2^n-1$个可能值的某个指定排序。那么，对于每个$r\geqslant1$有

$$\boldsymbol{P}^r=[P_{ij}^{(r)}]=[P(\Delta Y(r)=\alpha_j\mid\Delta X=\alpha_i)]\tag{8.32}$$

注意$\boldsymbol{P}$的每行元素的和为1，且由定理8.5.1知，均匀分布$\Pi=\left(\pi_1,\pi_2,\cdots,\pi_{\frac{1}{2^n-1}}\right)=\left(\frac{1}{2^n-1},\frac{1}{2^n-1},\cdots,\frac{1}{2^n-1}\right)$是其平稳分布，所以对每个$j$有

$$\frac{1}{2^n-1}=\pi_j=\sum_{i=1}^{2^n-1}\pi_ip_{ij}=\sum_{i=1}^{2^n-1}\frac{1}{2^n-1}p_{ij}=\frac{1}{2^n-1}\sum_{i=1}^{2^n-1}p_{ij}$$

进而$\sum\limits_{i=1}^{2^n-1}p_{ij}=1$，即$\boldsymbol{P}$的每列元素的和也为1。这样得到以下的结果。

定理 8.5.4 马尔柯夫密码的转移概率矩阵是双随机的，即每一行与每一列的和均为1。

8.6　马尔柯夫密码对差分密码分析的安全性

8.6.1　差分密码分析

差分密码分析是迄今已知对迭代密码最有效的分析方法之一，它是 1990 年由 Biham 和 Shamir 提出的，它对几乎所有的分组密码都适用，其基本思想是通过分析明文对的差值对密文对的差值的影响来恢复某些密钥比特。研究结果表明，迭代密码的简单轮函数 F 在如下意义下通常是密码上弱的：

对于 $Y_i=F(Y_{i-1},K_i)$ 和 $Y_i^*=F(Y_{i-1}^*,K_i)$，若三元组 $(\Delta Y_{i-1},Y_i,Y_i^*)$ 的一个或多个值是已知的，则确定子密钥 K_i 是容易的。

从而，若密文对已知，并且最后一轮的输入对的差分能以某种方式得到，则一般来说，确定最后一轮的子密钥或其一部分是可行的。在差分密码分析中，通过选择具有特定差分值 α_0 的明文对 (Y_0,Y_0^*)，使得最后一轮的输入差分 ΔY_{r-1} 以很高的概率取特定值 α_{r-1} 来达到这一点。

定义 8.6.1　i **轮差分**是一对差分 (α,β)，其中 α 是一对不同明文 X 和 X^* 的差分，β 是第 i 轮输出 Y_i 和 Y_i^* 的差分。一个 i **轮差分** (α,β) **的概率**是一条件概率，在明文 X 和子密钥 $K_1,\cdots,K_i$ 是独立、均匀随机时，明文差分 $\Delta X=\alpha$ 的条件下，第 i 轮输出的差分为 β 的概率。记为 $P\{\Delta Y(i)=\beta|\Delta X=\alpha\}$。

由式(8.32)可知，对于一个马尔柯夫密码，i 轮差分的概率是 i 步转移概率矩阵 $\boldsymbol{P}^i$ 的第 (α,β) 个元。

对 r 轮迭代密码的差分密码分析的基本过程可综述为以下的算法。

算法 8.6.1

第 1 步：找出一个 $(r-1)$ 轮差分 (α,β)，使得 $P\{\Delta Y(r-1)=\beta|\Delta X=\alpha\}$ 达到最大或几乎最大。

第 2 步：均匀随机地选择明文 X 并计算 X^*，使得 X 和 X^* 的差分为 α，找出 X 和 X^* 在实际密钥加密下所得的密文 $Y(r)$ 和 $Y^*(r)$。若最后一轮的子密钥 K_r（或 K_r 的部分比特）有 2^m 个可能值 $K_r^j(1\leqslant j\leqslant 2^m)$，设置相应的 2^m 个计数器 $\Lambda_j(1\leqslant j\leqslant 2^m)$，用每个 K_r^j 解密密文 $Y(r)$ 和 $Y^*(r)$，得到 $Y(r-1)$ 和 $Y^*(r-1)$，如果 $Y(r-1)$ 和 $Y^*(r-1)$ 的差分是 β，则给相应的计数器 Λ_j 加 1。

第 3 步：重复第 2 步，直到一个或几个计数器的值明显高于其他计数器的值，输出它们所对应的子密钥(或部分比特)。

在差分密码分析中，所有轮子密钥都是固定的，仅有明文能随机选择，而在计算差分概率时，明文和所有轮子密钥都是独立、均匀、随机的。因此在用计算得到的差分概率来确定哪一个差分在攻击中是有用的过程中，隐含了以下假设。

随机等价假设：对于所有实质上的 $r-1$ 轮高概率差分 (α,β)，式(8.33)，即

$$P\{\Delta Y(r-1)=\beta\mid\Delta X=\alpha\}$$

$$\approx P\{\Delta Y(r-1)=\beta \mid \Delta X=\alpha, K_1=k_1,\cdots,K_{r-1}=k_{r-1}\} \tag{8.33}$$

对子密钥值$(k_1,k_2,\cdots,k_{r-1})$的大部分成立。

若一个高概率差分对式(8.33)成立,则称它为差分密码攻击有用的(DC 有用的)。

对于分组长度为 n 比特的分组密码,存在 2^n-1 个 $\Delta Y(r-1)$的可能值,所以对于差分密码分析有以下结论。

定理 8.6.1 设随机等价假设成立,则一个具有独立子密钥的 r 轮迭代密码对差分密码分析是可破的当且仅当轮函数是弱的,且存在一个 DC 有用的 $r-1$ 轮差分(α,β),使得 $P\{\Delta Y(r-1)=\beta|\Delta X=\alpha\}\gg\frac{1}{2^n-1}$。

从上面的讨论可见,迭代分组密码抵抗差分密码分析的安全性依赖于差分概率的大小,更进一步,依赖于定义差分的群运算的选择。为使得差分密码分析更有效,群运算应该选择使得 DC 有用的差分概率达到最大。从现有的分析结果看,选择使分组密码称为马尔柯夫密码的群运算似乎是最恰当的。

从算法 8.6.1 可知,差分密码分析的数据复杂度两倍于选择明文对(X,X^*)的个数。差分密码分析的处理复杂度是从$(\Delta Y(r-1),Y(r),Y^*(r))$找出子密钥 K_r(或 K_r 的部分比特)的计算量,它实际上与 r 无关,而且由于轮函数是弱的,所以此计算量在大多数情况下相对较小。因此,差分密码分析的复杂度取决于它的数据复杂度。设 $C_d(r)$表示对 r 轮迭代密码进行差分密码分析的数据复杂度,下面的定理给出 $C_d(r)$的一个下界。

定理 8.6.2 设随机等价假设成立,则对 r 轮迭代分组密码进行差分密码分析的数据复杂度满足以下式子:

$$C_d(r)\geqslant\frac{2}{p_{\max}^{r-1}-\frac{1}{2^n-1}} \tag{8.34}$$

其中,$p_{\max}^{r-1}=\max_\alpha\max_\beta P\{\Delta Y(r-1)=\beta|\Delta X=\alpha\}$;$n$ 是分组长度。

证明:若差分密码分析成功,则 $\Delta Y(r-1)$预知 β 的次数至少比随机选择的 β' 的次数多一次,因此在 T 次试验内成功的必要条件是 $T\cdot p_{\max}^{r-1}\geqslant\frac{T}{2^n-1}+1$,其中一次试验是选择一对具有差分 α 的明文并加密这两个明文。数据复杂度 $C_d(r)=2T$,因此,$C_d(r)\geqslant\frac{2}{p_{\max}^{r-1}-\frac{1}{2^n-1}}$。

由不等式(8.34)可得,当 $p_{\max}^{r-1}\leqslant3\cdot2^{-n}$时,差分密码分析需要的数据几乎是所有的明文空间。因而,若 $p_{\max}^{r-1}\leqslant3\cdot2^{-n}$,就称此迭代分组密码对差分密码分析是实际安全的。

这里仅考虑使用一个差分进行差分密码分析的选择明文攻击,若预先知道更多高概率的差分,则差分密码分析有更有效的攻击算法。设 A 是攻击者预先知道的明文差分集合,对于每个 $\alpha\in A$,都存在一个高概率的 $r-1$ 轮差分(α,β_α)。对于每对明

密文对$(X,Y(r))$和$(X^*,Y^*(r))$，如果 X 和 X^* 的差分是 $\alpha\in A$，则利用差分(α,β_α)找出子密钥 K_r(或 K_r 的部分比特)的所有可能值，并在相应的计数器上加 1。对每对这样的明文对重复上述操作，若子密钥 K_r(或 K_r 的部分比特)的某些值的计数明显高于别的值的计数，则这些值被当作实际子密钥 K_r 的可能值，并用其他的方法对它们做进一步检测。

假设用一个差分的差分密码分析需要 T 个选择明文，$|A|=N$，且 N 个差分具有大约相同的概率，则上述攻击方法所需的选择明文大约为$\dfrac{T}{\sqrt{N}}$个。

最早的差分密码分析用特征的概念，i **轮特征**是一个 $i+1$ 元组$(\alpha,\beta_1,\cdots,\beta_r)$，它被看成是$(\Delta X,\Delta Y(1),\cdots,\Delta Y(i))$的一个可能值。1 轮特征和 1 轮差分是一样的，而 i 轮特征只是 i 轮差分的一条序列。i **轮特征的概率**定义为

$$P\{\Delta Y(1)=\beta_1,\Delta Y(2)=\beta_2,\cdots,\Delta Y(i)=\beta_i\mid\Delta X=\alpha\}$$

这里明文 X 和子密钥 $K_1,K_2,\cdots,K_i$ 都是独立、均匀、随机的。

由于在对 r 轮迭代分组密码进行差分密码分析时，为了确定子密钥 K_r 仅需要知道 $\Delta Y(r-1)$，而不管中间差分 $\Delta Y(j)(1\leqslant j<r-1)$是什么，所以使用差分(和差分概率)比特征(和特征概率)能更好地刻画差分密码分析的成功概率。不过在分析中使用特征和特征概率更实用，原因是 i 轮特征的概率可由 1 轮特征的概率容易计算出。

对于具有独立均匀随机的轮子密钥的马尔柯夫密码，r 轮特征$(\beta_0,\beta_1,\cdots,\beta_r)$的概率由下面的 C-K 方程给出：

$$P\{\Delta Y(1)=\beta_1,\Delta Y(2)=\beta_2,\cdots,\Delta Y(i)=\beta_r\mid\Delta X=\beta_0\}$$
$$=\prod_{i=1}^{r}P\{\Delta Y(1)=\beta_i\mid\Delta X=\beta_{i-1}\}$$

由等式(8.33)可得，r 轮差分(β_0,β_r)的概率是

$$P\{\Delta Y(r)=\beta_r\mid\Delta X=\beta_0\}$$
$$=\sum_{\beta_1}\sum_{\beta_2}\cdots\sum_{\beta_{n-1}}\prod_{i=1}^{r}P\{\Delta Y(1)=\beta_i\mid\Delta X=\beta_{i-1}\}$$

这里和式是所有可能的非幺元求和。

对于迭代分组密码的差分密码分析，确定差分的概率是很重要的。对于马尔柯夫密码，这种概率由它的转移矩阵唯一确定。值得说明的是，在对 r 轮迭代分组密码进行差分密码分析时，有时并不需要 $r-1$ 轮的高概率差分，可能更短轮的高概率差分就可以满足攻击需求，这要依据具体的分组密码而分析。

8.6.2　马尔柯夫密码的安全性

迭代分组密码的安全性是基于：一个密码上的“强”函数可通过迭代密码上的“弱”函数足够多次来得到。下面的结果表明，满足某些特性的马尔柯夫密码，迭代将产生能够抵抗差分密码分析的安全密码。

定理 8.6.3　对于具有独立、均匀、随机子密钥的分组长度为 n 的马尔柯夫密

码，若马尔柯夫链 $\Delta X=\Delta Y(0),\Delta Y(1),\cdots$ 具有遍历性，则它的极限分布必定是均匀分布，即对任意非幺元差分(α,β)式(8.35)成立，即

$$\lim_{r\to\infty}P\{\Delta Y(r)=\beta\mid\Delta X=\alpha\}=\frac{1}{2^n-1}\tag{8.35}$$

若再假设随机等价假设成立，则经过充分多的迭代，该马尔柯夫密码能抵抗差分密码分析。

证明：由定理 8.4.1 可知具有遍历性的有限齐次马尔柯夫链具有唯一的极限分布，而且极限分布也是它的平稳分布。又由定理 8.5.1 可知它的平稳分布是均匀分布，所以对于任意非幺元差分(α,β)式(8.35)成立。

由定理 8.4.1 和定理 8.4.2 可知，若马尔柯夫密码的差分链是不可约的和非周期的，则它一定满足式(8.35)。差分链是非周期的一个充分条件是存在 α，使得形如(α,α)的 1 轮差分具有非 0 概率。对于差分链是不可约性有以下的结论。

定理 8.6.4 对于定理 8.5.2 所示的马尔柯夫密码，差分链是不可约的当且仅当对每个明文对(X,X^*)及每个密文对(Y,Y^*)，都存在整数 r_0 和 r_0 个子密钥的一种选择，使得 r_0 轮加密将 X 加密为 Y，将 X^* 加密为 Y^*。

证明：轮函数具有形式：

$$f(X,Z)=g(X\otimes Z_A,Z_B)$$

其中$\otimes$是定义差分的群运算。令 $S=X\otimes Z_A^{(1)}$ 和 $S^*=X^*\otimes Z_A^{(1)}$，则 $\Delta S=\Delta X$。

假设差分链是不可约的，对于给定的明文对(X,X^*)和密文对(Y,Y^*)，令 $\alpha=X\otimes X^*$，$\beta=Y\otimes Y^*$，由不可约性可知，存在 r_0 使得

$$P\{\Delta Y(r_0)=\beta\mid\Delta S=\alpha\}=P\{\Delta Y(r_0)=\beta\mid\Delta X=\alpha\}>0$$

它意味着存在子密钥 $Z_B^{(1)},Z_A^{(2)},Z_B^{(2)},\cdots,Z_A^{(r_0)},Z_B^{(r_0)}$ 和差分为 α 的 S 和 S^*，使得在这些密钥下，S 被加密为 Y、S^* 被加密为 Y^*。令 $Z_A^{(1)}=X^{-1}\otimes S$，由于 $\Delta S=\Delta X$，所以 $S^*=X^*\otimes Z_A^{(1)}$。因此，在所选子密钥 $Z_A^{(1)},Z_B^{(1)},Z_A^{(2)},Z_B^{(2)},\cdots,Z_A^{(r_0)},Z_B^{(r_0)}$ 下，明文对(X,X^*)将产生密文对(Y,Y^*)。

逆是显然的，且事实上对所有马尔柯夫密码均成立。

上面的结论显示，如果马尔柯夫密码的差分链是不可约的和非周期的，则此密码经过充分多轮迭代将能抵抗差分密码分析，但在密码算法设计中，由于实现性能的因素，往往对迭代轮数有限制。下面利用转移矩阵的特征值给出马尔柯夫密码抵抗差分密码分析所应迭代的轮数。

设 $\boldsymbol{P}$ 是马尔柯夫密码的转移矩阵，矩阵 $\boldsymbol{P}$ 的特征值是一个实数 λ，关于它存在一个向量(称为 λ 的特征向量)$\boldsymbol{V}=(v_1,v_2,\cdots,v_{2^n-1})$使得 $\boldsymbol{VP}=\lambda\boldsymbol{V}$。从定理 8.4.1 可知，1 是矩阵 $\boldsymbol{P}$ 的一个特征值，并且$\left(\frac{1}{2^n-1},\cdots,\frac{1}{2^n-1}\right)$是其特征向量，由关于非负矩阵的 Perron-Frobenius 定理可知，若 $\boldsymbol{P}$ 是本原的，则 $\boldsymbol{P}$ 的其余特征值的模严格小于 1，并且当 r 趋于无限时，$P_{ij}(r)-\frac{1}{2^n-1}$随着 r 指数地趋于 0。假如具有第二大模的特征值 λ_2 的重数是 t_0，且设模较小的特征值的重数至多为 t_0，则存在一个常量 $a>0$ 使得

对于所有差分对(α,β)均有

$$P\{\Delta Y(r)=\beta\mid\Delta X=\alpha\}-\frac{1}{2^n-1}\leqslant ar^{t_0-1}\mid\lambda_2\mid^r \tag{8.36}$$

进一步，利用定理 8.6.2 和数据复杂度的上界为 2^n 可得，若 r_0 是使得

$$ar_0^{t_0-1}\mid\lambda_2\mid^{r_0}\leqslant\frac{1}{2^n} \tag{8.37}$$

的最小整数，则对所有 $r>r_0$，经过 r 轮迭代后，马尔柯夫密码实际上能抵抗差分密码分析。

8.7　注记

本章重点介绍了一些在信息安全尤其是在密码学研究中常用的随机过程方法与技术，并以马尔柯夫密码为例介绍了随机过程方法与技术在密码学中的应用。其目的是为了满足信息安全领域中的基本应用而选材的，对随机过程感兴趣的读者可进一步参阅文献[4～6]。

参 考 文 献

[1] Biham E, Shamir A. Differential Crytanalysis of DES-like Cryptosystems. Journal of Cryptology, Vol. 4, No. 1, pp. 3-72. 1991

[2] Lai X, Massey J L. Markov Ciphers and Differential Cryptanalysis. Advances in Cryptology-Eurocrypt'91, LNCS 547, pp. 17-38, Springer-Verlag, Berlin 1991

[3] Lai X. On the design and security of block ciphers. ETH Series in Information Processing, (Edt: Massey J L), Vol. 1, Hartung-Gorre Verlag, Konstanz, 1992

[4] 盛骤，谢式千，潘承毅. 概率论与数理统计. 北京：高等教育出版社，1989

[5] 方兆本，缪柏其. 随机过程. 北京：科学出版社，2004

[6] 叶尔骅，张德平. 概率论与随机过程. 北京：科学出版社，2005

[7] 廖安平，刘建州. 矩阵论. 长沙：湖南大学出版社，2005

第 9 章　信息论方法与技术

1948 年，C. E. Shannon 在贝尔实验室技术期刊上发表了题为“通信的数学理论”一文[1]，从此奠定了信息论的理论基础。次年，即在 1949 年，Shannon 又在同一期刊上发表了“保密系统的通信理论”一文[2]，于是又奠定了对现代密码学的系统研究的理论基础。Shannon 的这两篇论文开辟了现代通信进而保密通信的科学研究方向。

Shannon 在“通信的数学理论”一文中，通过量化度量精确描述了事件的信息、熵和事件集合（信源）的信息和熵，以及两个事件（信源）之间的条件信息、互信息等。通过这些量化描述，给出了信源编码定理、信道容量定理和信道编码定理。信息论方法对通信中信源编码和信道编码的指导意义是十分重要的。除此之外，信息论方法对理论密码学的研究也有着重要的指导作用。不同于它在信源和信道编码中的直接指导作用，信息论方法对密码学的指导仅限于理论保密程度的度量和描述，而理论保密与实际保密之间又有着非常微妙的关系：许多实际使用的密码体制不具有理论保密性，即在密文中包含了原始消息（明文）的全部信息。它之所以能够在实际中使用，是因为要想从密文中提取明文信息，所花的计算代价无法承受，即许多实际中使用的密码方案是计算安全的。另一方面，一些理论保密的体制，如 Shannon 提出的一次一密体制，在实际中使用的代价太大，因为密钥协商和分发是影响其应用的关键困难，特别是在广泛的商业应用中更是如此。当然，实际中也有一些特殊的密码体制是理论保密的，当然它们更是实际保密的，即使攻击者将来在计算能力方面有很大提高，对理论保密的体制则没有任何威胁。

本章主要介绍了信息论的基本概念和性质，同时讨论了信息论方法在信息安全尤其是在密码设计与分析中的应用。

9.1　事件的信息度量

通信中人们关心的是信源（也称为发信人）和信宿（也称为收信人）之间的消息传递。由于信道可能造成的错误，使得收信人所收到的消息可能与发信人所传送的消息不同，通常以很小的错误概率发生。而通信的过程就是收信人根据所接收到的消息来恢复发信人所传送的消息的过程。可以把信源和信宿看作两个随机变量，信源输出的符号是这个随机变量的一个特殊取值，同样信宿所接收到的符号也是信宿这个随机变量的一个取值。这样，便可以通过随机变量来研究信源和信宿这两个通信中的主体。

一个随机变量 X 是一个集合，其中的任何一个元素都是这个随机变量以某种概率的可能取值，所有这些概率的和为 1，因为当该随机变量输出一个符号时，它一定是这个集合中的符号。因此描述随机变量不能简单地用集合表示，而是具有概率分布的集合元素。

下面考虑什么是信息和如何度量信息。从直观上讲,信息的定义很难恰当地给出。在现实中,信息是对所给定的某个事物(一个随机变量的某个特定取值或状态)的全部理解的总和。如看到某人发愁的面孔,你可能会猜测到某些东西,这些猜测就是那个人的面孔所提供给你的信息。但这并不是信息的全部,因为你提取信息的能力可能有限,另外一些人看到这张面孔可能会得到另外的信息。这些信息的总和就是这张面孔所提供的信息。但是,对于这样的信息,人们很难给出定量的度量标准,更难确定它们的总和。幸运的是,通信中所处理的消息远没有人的面孔这么复杂,仅仅是简单的有限集合的符号。更特别地,考虑的随机变量可能只取 0 和 1 这两个值,这样一来,就容易给出信息的度量。

定义 9.1.1　假定某随机变量 X 的所有可能取值为 x_i, $i=1,2,\cdots,n$,而事件 $X=x_i$ 的概率为 $P(x_i)$。则事件 $X=x_i$ 的自信息(self-information)定义为

$$I(x_i)=\log_2\left(\frac{1}{P(x_i)}\right)=-\log_2 P(x_i) \tag{9.1}$$

根据定义 9.1.1,可以注意到下列问题:

(1) 事件的概率越低,事件发生时所带来的信息越多;

(2) 计算自信息的公式中没有给出对数的底数。事实上,不同底数的对数之间只相差一个常数,因此原则上自信息可以使用任何底数计算,只不过所得结果的单位不同。通常取 2 为底数的对数,此时所得结果的单位为“比特”(bit)。

从直观上来看,也能体会到小概率事件带来更多信息这一特征。这也是为什么一些很离奇(因而发生的概率很小)的事情一旦发生,就在很长一段时间内被人们议论,而一些很平常的事情(因而发生的概率很大),可能都不会引起别人的注意。下面给出两个小例子计算事件的信息量。

例 9.1.1　投掷硬币可能出现两种结果:正面朝上(即事件 $X=x_0$ 发生),或反面朝上(即事件 $X=x_1$ 发生)。正常情况下有 $P(x_0)=P(x_1)=0.5$,因此该信源的每一个输出所含的信息量为

$$I(x_i)=-\log_2 P(x_i)=-\log_2(0.5)=1(\text{bit}),\quad i=0,1 \tag{9.2}$$

上述例子说明,在投掷硬币时,每一次投掷结果所带来的信息量都是 1 比特。那么对投掷多次的情况如何呢?假定每一次投掷都是独立的事件,即任何已经投掷的结果对即将进行的投掷结果没有任何影响,而且事件所发生的概率保持不变。

例 9.1.2　信源为连续投掷硬币 m 次的输出结果。用 0 表示正面朝上,1 表示正面朝下,x_i 表示第 i 次的投掷结果。则将输出结果看作随机事件 X 时,X 的状态可以为 $x_1,x_2,\cdots,x_m$ 的任意值,而且等可能地发生。因此每一个输出结果所携带的信息为

$$\begin{aligned}I(x_1,x_2,\cdots,x_m)&=-\log_2 P(x_1,x_2,\cdots,x_m)\\&=-\log_2\prod_{i=1}^{m}P(x_i)=-\log_2(2^{-m})=m(\text{bit})\end{aligned} \tag{9.3}$$

有时,事件之间不是独立的,而是相互影响的,包括不同随机变量的状态。如喜欢喝酒的人可能比不喝酒的人更容易吸烟,因此喝酒和抽烟这两个随机变量之间存

在着某种微妙的联系，这种联系虽然不能很确定它意味着什么，但却以一定概率发生影响。数据挖掘的目的之一就是发现不同事件（不同随机变量的状态）之间的联系并加以利用。下面所考虑的不是同一随机变量不同状态之间的联系，而是不同随机变量之间的联系，并假定这些随机变量分别是无记忆的，即各自的不同状态之间没有任何联系或影响。假定 X 和 Y 是两个随机变量，它们的状态分别取自有限集合 $\{x_1, x_2, \cdots, x_m\}$ 和 $\{y_1, y_2, \cdots, y_n\}$。首先给出互信息的定义。

定义 9.1.2 定义事件 x_i 和 y_j 之间的互信息 $I(x_i;y_j)$ 为

$$I(x_i;y_j) = \log_2\left(\frac{P(x_i \mid y_j)}{P(x_i)}\right) \tag{9.4}$$

其中 $P(x_i \mid y_j)$ 为事件 y_j 发生条件下，事件 x_i 发生的概率。特别地，当满足 $P(x_i|y_j)=P(x_i)$，即事件 y_j 的发生不影响事件 x_i 发生的概率时，称事件 x_i 与 y_j 为独立事件。

因为在事件 y_j 发生的条件下，事件 x_i 发生的概率可能会变大，也可能会变小，因此互信息 $I(x_i;y_j)$ 的值可能为负值。当 x_i 与 y_j 为独立事件时，它们之间的互信息为 0。

上述互信息的定义考虑的是事件 y_j 发生条件下，事件 x_i 发生的概率。为什么一定要事件 y_j 先发生，而事件 x_i 后发生呢？如果它们发生的次序刚好相反会怎么样呢？事实上，互信息的定义与事件发生的次序无关。根据概率等式 $P(A,B)=P(A)P(B|A)=P(B)P(A|B)$ 可以得到

$$\frac{P(x_i \mid y_j)}{P(x_i)} = \frac{P(x_i \mid y_j)P(y_j)}{P(x_i)P(y_j)} = \frac{P(x_i, y_j)}{P(x_i)P(y_j)} = \frac{P(y_j \mid x_i)}{P(y_j)}$$

因此有

$$I(x_i;y_j) = \log_2\left(\frac{P(x_i \mid y_j)}{P(x_i)}\right) = \log_2\left(\frac{P(y_j \mid x_i)}{P(y_j)}\right) = I(y_j;x_i) \tag{9.5}$$

式(9.5)说明互信息的定义与事件的次序无关，这也符合我们的直觉，即事件 x_i 的发生为事件 y_j 的发生所提供的信息等于事件 y_j 的发生为事件 x_i 的发生所提供的信息。下面考虑两种特殊情况。

假设事件 x_i 与事件 y_j 独立，即有 $P(x_i|y_j)=P(x_i)$，则

$$I(x_i;y_j) = \log_2\left(\frac{P(x_i \mid y_j)}{P(x_i)}\right) = \log_2(1) = 0$$

这再一次认证了独立事件相互之间不提供信息这一事实。

假设事件 y_j 的发生完全确定事件 x_i 的发生，即 $P(x_i|y_j)=1$，则有

$$I(x_i;y_j) = \log_2\left(\frac{P(x_i \mid y_j)}{P(x_i)}\right) = \log_2\left(\frac{1}{P(x_i)}\right) = I(x_i)$$

这刚好是事件 x_i 的自信息。这可以理解为当事件 y_j 发生时，它提供了关于事件 x_i 的全部信息。

下面考虑事件的条件自信息。首先给出定义。

定义 9.1.3 在事件 y_j 发生的前提下，关于事件 x_i 的自信息称为条件自信息(conditional self-information)，表示为

$$I(x_i \mid y_j) = \log_2\left(\frac{1}{P(x_i \mid y_j)}\right) = -\log_2 P(x_i \mid y_j) \tag{9.6}$$

根据定义 9.1.3,可以将互信息写为

$$\begin{aligned} I(x_i;y_j) &= \log_2\left(\frac{P(x_i \mid y_j)}{P(x_i)}\right) = \log_2 P(y_j \mid x_i) - \log_2 P(x_i) \\ &= I(x_i) - I(x_i \mid y_j) \end{aligned} \tag{9.7}$$

上述式(9.7)给出的是互信息、自信息和条件自信息之间的联系。

9.2 随机变量的信息度量

上一节讨论的是随机变量的某些特定的状态(称为值或事件)的信息度量,单独的和相互的。作为这些所有可能事件全体的随机变量,也有一种类似的对信息的度量。为使讨论问题简单,假定所讨论的随机变量的状态(或这些事件)相互独立,即一个状态的出现不影响任何另一个状态的出现。

定义 9.2.1 假设随机变量 X 和 Y 所有可能的状态分别来自有限集合 $\{x_1, x_2, \cdots, x_n\}$ 和 $\{y_1, y_2, \cdots, y_m\}$。则 X 与 Y 之间的平均互信息定义为

$$\begin{aligned} I(X;Y) &= \sum_{i=1}^{n}\sum_{j=1}^{m} P(x_i, y_j) I(x_i; y_j) \\ &= \sum_{i=1}^{n}\sum_{j=1}^{m} P(x_i, y_j) \log_2\left(\frac{P(x_i, y_j)}{P(x_i)P(y_j)}\right) \end{aligned} \tag{9.8}$$

上述随机变量的互信息的意义可以理解为所有可能的事件对 (x_i, y_j) 的互信息的加权平均,而权重就是这两个事件共同发生的概率。容易证明下述性质成立。

性质 1: $I(X;Y) = I(Y;X) \geqslant 0$。

性质 2: 当且仅当 X 与 Y 统计独立时,有 $I(X;Y) = I(Y;X) = 0$。

值得注意的是,尽管事件的互信息可能为负值,但随机变量的互信息不可能为负值。下面给出随机变量平均自信息的定义。

定义 9.2.2 随机变量 X 的平均自信息为

$$H(X) = \sum_{i=1}^{n} P(x_i) I(x_i) = -\sum_{i=1}^{n} P(x_i) \log_2 P(x_i) \tag{9.9}$$

注意上述对随机变量平均自信息的表示符号为 $H(X)$,而不是类似于事件的自信息,是因为这样定义的平均自信息有着特别的意义,它表示一个系统的不确定程度,而在统计动力学中表示系统不确定程度的量称为系统的“熵”(entropy)。因此这里采用同样的符号,而且也称 $H(X)$ 为随机变量 X 的熵。

类似地,把条件平均自信息又称为条件熵,定义如下。

定义 9.2.3 随机变量 X 对于 Y 的条件平均自信息或条件熵定义为

$$I(X \mid Y) = \sum_{i=1}^{n}\sum_{j=1}^{m} P(x_i, y_j) \log_2 \frac{1}{P(x_i \mid y_j)} \tag{9.10}$$

根据上面所给出的关于平均互信息、平均自信息和条件平均自信息的定义不难得到下述等式,即

$$I(X;Y) = H(X) - H(X \mid Y) = H(Y) - H(Y \mid X) \tag{9.11}$$

由上述性质 1 可得到 $H(X) \geqslant H(X|Y)$。对此不等式的解释是：对 Y 的观察不会增加 X 的不确定性(熵)。注意这与对个别事件的情况是不同的，因为对 Y 的某个事件 y_j 的观察结果可能会增加对 X 的某个事件 x_i 的错误判断。但是对于一个随机变量来说，对其他随机变量的观察不会增加其不确定性。当这两个随机变量独立时，对一个随机变量的观察也不会减少另一个随机变量的不确定性。

9.3 信源编码定理

度量信息的目的之一在于高效率地用数字表示信息。这种用数字或符号串表示信息的方法就是信源编码。下面通过一个小例子来说明什么是高效率的信源编码。假定要表达的信息是 3 种天气情况，即信源的 3 种不同消息分别为"晴天"、"阴天"和"下雨"。当然分别用这些对应的中文字表示信源的不同消息也是一种信源编码方式，但效率非常低下。我们知道，计算机在存储中文字时，每一个中文字至少要占用两个字节，即 16 比特的存储空间。这样表示全部上述信源信息需要 6 个中文字，即 96 比特的存储空间。通过观察发现可以将中文表示分别简化为"晴"、"阴"和"雨"，而不会发生意义含糊的情况，这样一下子将编码效率提高了 1 倍，即只用 48 比特的存储空间就可以记录信源的全部消息。但是这仍然不够理想。更简单的方法是将信源消息编号，将每个编号对应一个编码。对上述 3 个消息的信源，需要对编号 1、2、3 进行编码，可以分别用 01、10 和 11 表示，即它们对应的二进制表示。这似乎已经很有效了，因为只用了 6 比特就可以表示全部信源消息。是否还可以找到更高效率的编码方式呢？即是否能使用少于 6 比特的存储空间来表示所有消息？下面简单地介绍信源编码，这个问题的答案可以从下面的介绍中得到。

信源编码的目的在于用最少的数据资源表示出所期望的信息，这样一来，无论在通信还是数据存储中，只需要处理这些编码即可。当需要恢复信息时，对从存储介质中获取的或通信信道中接收到的码子序列对应到原来的实际信源信息即可。当然编码的规则对于信息恢复者来说应该是公开的，因为这里还不涉及信息保密问题。为了能从码子序列中正确地恢复原始消息，编码规则应满足下述性质：①从码子序列到原始消息的对应(即译码)有明确的规则；②在码子不发生错误的情况下，由码子序列可以正确恢复原始消息(即正确译码)。如果还考虑码的性能，则要求表示同样的信息，不能用更短的数据(码子序列)表示，即所得到的码是最优的。

9.3.1 定长信源编码

假定信源有 n 个符号，将它们依次赋予序号 $0,1,2,\cdots,n-1$，然后对每个序号用等长的二进制表示，则这些表示的全体就是一种编码方案。为了使编码效率最高，必须要求这些二进制表示的长度最短。对于 n 个符号，容易证明编码长度不得小于 $k=\lceil \log_2 n \rceil$，其中$\lceil X \rceil$表示不小于 X 的最小整数。这种将所有信源消息都编码为固定长度码子的编码方式称为定长信源编码。

定理 9.3.1　假定信源 X 有 n 个符号。将 n 个符号的序号 $0,1,2,\cdots,n-1$ 依次表示为长度为 $k=\lceil \log_2 n \rceil$的二进制表示，是一种最优的等长信源编码。

证明：需要证明定理 9.3.1 中给出的编码规则满足上述最优编码的性质。首先不难验证，译码规则是将所获得的码子序列按照 k 比特截段，然后将每一段对应到原始消息。这种明确的译码规则将每一个原始信源消息对应到长度为 k 的一个二元数组，在码子不发生错误的情况下，显然逆过程也正确，即每一个编码中的二元数组对应唯一的信源消息。另外很显然，不能用少于 k 的二元数组来表示信源消息，因为所有这样的固定长度的二元数组的个数少于信源消息的个数，即对于所有 $t<k$，有 $2^t<n$。这就证明了上述码子的最优性。

当 $n=2^k$ 时，上述编码用到所有的长度为 k 的二元数组。但是当 $n<2^k$ 时，只用到部分二元数组。如果不考虑编码、译码过程的复杂性，只考虑正确性的话，在此情况下可以用部分剩余的二元数组替换原来的一些码子，于是得到另外一种编码方式，而且容易验证，这种新的编码方式仍然是最优的等长信源编码。不难看出，这些经替换得到的编码共有 $\binom{2^k}{n}$ 个，包括原始编码。

9.3.2　变长信源编码

回过头来再看一下本节一开始的例子，即信源消息共有 3 个符号“晴”、“阴”和“雨”。通过对它们的标号 0、1、2 编码，得到定长编码的码子分别为 00、01、10。这是不是最优的码子呢？根据定理 9.4.1，对于定长编码，它们是最短的。但是，如果打破定长的限制，可能会得到不同的结果。比如将前两个码子分别变为 0 和 1，这样总的码长缩短了，码的性能也提高了。接下来的问题是能否唯一译码，即当收到一个码子序列时，假定没有发生错误，能否唯一恢复到原始消息。很遗憾，当要恢复的码子序列是 10 时，它有可能表示“雨”，也有可能表示“阴”和“晴”，即不能唯一译码，因此这样的简化不可用或没意义。如果只将第一个码子变为 0，而其余不变呢？当对码子序列 01010 进行译码时，可能将其分段为 0 10 10，然后译为“晴”、“雨”、“雨”，或者将其分段为 01 01 0，然后译为“阴”、“阴”、“晴”。看来这种简化也是不可行的。是否存在一种可行的方式将码子总长度变短呢？再看另一种方式：保持前两个码子不变，将最后一个码子 10 变为 1。对于这个小例子，容易看出可以唯一恢复原始消息。事实上，只要对码子序列按次序查看编码表，只要得到一个合法码子就进行译码，这样便不会发生错误。这个例子说明，变长编码可以比定长编码变得更高效，当然编码过程和译码过程可能比定长编码更为复杂。

上述例子通过修改定长编码来得到变长编码时之所以能或不能唯一译码，关键在于是否存在某些码子是另外一些码子的前缀。在第一种修改中，01 变成了 1，但 1 是最后码子 10 的前缀，因此不能唯一译码。对第二种修改，00 变成了 0，但 0 是第二个码子 01 的前缀，因此仍然不能唯一译码。而第三种修改将 10 变为 1，但 1 不是任何其他两个码子的前缀，因此可以正确译码。这种没有一个码子是另一码子前缀的信源码称为前缀码。如果将所有码子逆向写，则前缀码就变成了后缀码，它们具有同

样的性质(唯一可译性和码的效率等)。为此,只需要讨论前缀码即可。

对于一个前缀码,假设各信源符号的编码长度依次为 $L_1 \leqslant L_2 \leqslant \cdots \leqslant L_n$,则一定满足下述不等式(称为 Kraft 不等式),即

$$\sum_{i=1}^{n} 2^{-L_i} \leqslant 1 \tag{9.12}$$

变长信源编码的更多优势在于当信源消息的概率分布已知时。对定长信源编码来说,信源消息的概率分布没有任何影响,这样大概率消息和小概率消息占有相同长度的码子,因此会造成浪费。而变长信源编码可以将信源消息的概率分布充分利用,使得信源输出消息对应的码子序列平均长度最短。下面介绍一种著名的变长信源编码算法——Huffman 编码。

算法 9.3.1 (Huffman 信源编码算法)

假设信源符号 x_i 的出现概率为 $P(x_i)$,$i=1,2,\cdots,n$,则编码步骤如下。

(1) 将信源符号按概率递减次序由上至下排列。

(2) 将最下面的两个符号连接在一起,并分别对两个分支标记为 0 和 1。将它们的概率和作为一个新符号的概率。重新选取最小两个概率的符号,重复上述连接编码方法。

(3) 将此过程进行下去,直到只剩下一个概率(应该等于 1),这就完成了 Huffman 树的构造。

对任何符号,找到其到最后节点的一个路径,再沿反方向记下分支标号,就是该符号的码子。

下面给出一个例子说明 Huffman 编码是如何进行的。

例 9.3.1 假设信源 X 有 7 个符号 $x_1,x_2,\cdots,x_7$,它们的概率分别为 0.37、0.33、0.16、0.07、0.04、0.02、0.01。首先根据算法 9.3.1 建立 Huffman 树,如图 9.1 所示。

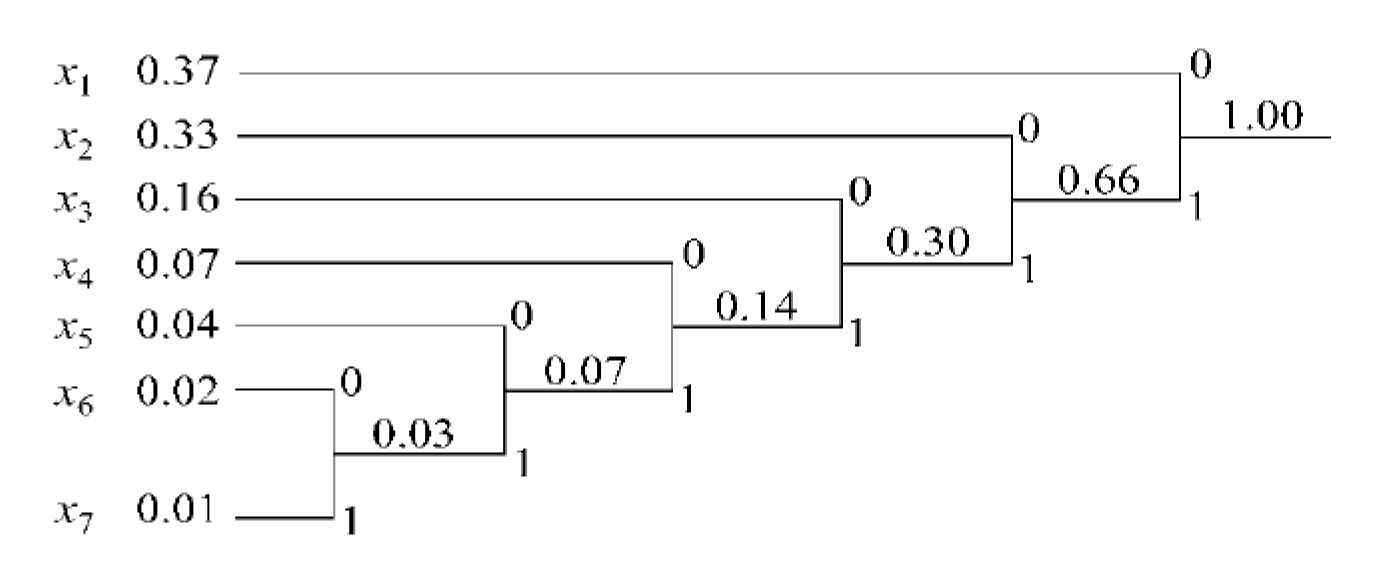


图 9.1 Huffman 树

根据 Huffman 编码算法的第(2)步,沿 Huffman 树的最后节点到每个符号的路径的标号就是该符号的码子,于是得到信源符号 x_1、x_2、$\cdots$、x_7 对应的码子分别为 0、10、110、1110、11110、111110、111111。这个看上去不怎么好的编码,对于这个特定概率分布的信源,却是最优的。

由于 Huffman 编码过程中给 Huffman 树的任何两个分支分配标记符号 0 和 1

时是随机的,因此 Huffman 编码不是唯一的,但它们的性能是相同的,即每个符号对应的码子的长度是相同的,而且都是前缀码。

9.3.3 信源编码定理

既然信源编码的目的是用尽量少的比特数来表示信源消息(或信源符号),当信源各符号出现概率不同时,应该考虑在一定编码方式下,每个信源符号平均所占用的码子长度(比特数)。假定信源符号 x_i 的出现概率为 $P(x_i)$,该符号的码子长度为 $L(x_i)$,则信源符号平均所占的比特数为

$$\overline{R} = \sum_{i=1}^{n} L(x_i)P(x_i) \tag{9.13}$$

Shannon 从理论上给出了对离散无记忆信源的变长编码的平均码长所能达到的理论区间,即下面著名的信源编码定理。

定理 9.3.2(信源编码定理) 设 X 为离散无记忆信源(DMS)的字母集合,其有限熵为 $H(X)$,而其输出符号 x_i 的出现概率为 $P(x_i)$,$i=1,2,\cdots,n$,则存在满足前缀条件的码,其平均码长满足不等式

$$H(X) \leqslant \overline{R} \leqslant H(X)+1 \tag{9.14}$$

上述信源编码定理说明:用来表示信源符号的平均最小比特数必须至少等于该信源的熵。该定理同时也说明,一个高熵(不确定性)的信源必须用更多比特数来表示信源符号。

前面 Huffman 编码的信源符号集合为$\{x_1,x_2,\cdots,x_7\}$,各符号的出现概率分别为 0.37、0.33、0.16、0.07、0.04、0.02、0.01。则信源熵为 $H(X) = -\sum_{i=1}^{7} P(x_i)\log_2 P(x_i) = 2.1152(\text{bit})$。而通过 Huffman 编码后的平均每个符号的码子长度(比特数)为

$$\begin{aligned}\overline{R} &= \sum_{i=1}^{7} L(x_i)P(x_i) = 1(0.37)+2(0.33)+3(0.16) \\ &\quad +4(0.07)+5(0.04)+6(0.02)+6(0.01) \\ &= 2.1700(\text{bit})\end{aligned}$$

信源编码定理说明,任何有效编码的平均码子长度不小于信源熵。平均码子长度越接近信源熵,说明编码效率越高。定义**编码效率**为信源熵与平均码子长度的比值(商),即 $\eta=H(X)/\overline{R}$。上例的编码效率为 $\eta=2.1152/2.17=0.9747$。虽然这个值小于 1,但已经很接近理论最优值了。而在实际中,按照单个信源符号的信源编码不可能再提高编码效率,除非考虑对信源符号进行组合的编码方式。对这种方法这里就不再介绍了。

9.4 密码体制的理论安全性

所谓密码体制,是这样一个系统,它包括明文空间,即所有可能的被加密消息的集合 M,密文空间,即所有可能的明文加密后的消息的集合 C,和从 M 到 C 的一些一

对一映射的集合。这些映射由一个称为密钥的参数控制，因此密码体制的另一重要元件是密钥空间，即所有可能的密钥的集合 K。每一个密钥确定从 M 到 C 的一个一对一映射，而且由这个密钥还可以唯一确定该映射的逆映射，即从 C 的某个子集到 M 的一对一映射。这些由密钥确定的映射有时被描述为从 M 到 C 的一个一对一映射，只不过在实际确定映射时需要密钥来调整该映射，使得同一消息在不同密钥下对应的像可能不同。习惯上，把消息集合 M 中的元素称为明文，把加密后的消息集合 C 中的元素称为密文，把这个从 M 到 C 的映射称为加密算法 E，而参与到加密过程中的 K 中的某个元素称为加密密钥。当加密密钥 $k\in K$ 确定后，从 M 到 C 的加密映射也就确定了。习惯上，把这一特定的映射记为 $c=E_k(m)$，这就是对 M 中消息的一个具体加密算法。对应地，解密算法是从 C 到 M 的一个映射，严格地说是从 C 的某个子集合到 M 的映射，记为 $m=D_k(c)$，这就是对应的解密算法。因为解密算法是加密算法的逆过程，因此有时为比较方便，也将解密算法记为 $m=E_k^{-1}(c)$。

密码体制的安全性在于当敌手获取到部分密文时，是否有能力得到它们对应的原始消息。在实际应用中，从密文恢复原始明文需要大量计算，因此计算能力也是衡量密码体制安全性的一个重要标准。但是，敌手的计算能力是随时间和科技进步的发展而发生变化的，如果敌手在计算能力上发生突变，则建立在计算复杂性假设上的密码体制可能会变得不再安全。有时为了讨论一个密码体制的理论安全性，假定敌手或密码分析者具有无限的计算能力。这个假设虽然不很实际，但敌手的计算能力以不确定的速度发展这一事实可以从某种程度上说明假设是合理的。另外还要假设敌手或密码分析者知道加密算法 E，但不知道加密过程中使用的密钥。这一假设在早期密码工具被军事垄断的时代被认为不很合理，因为没有谁愿意把自己设计的密码算法泄漏给敌手。但在今天的电子商务安全方面，这一假设是必要的，而且许多事实表明，通过保护加密算法来增加系统的安全性是很难成功的。一个很具有代表性的例子就是用于 GSM 手机系统的 A3 和 A5 加密算法，它们本来被设计成不公开的算法，但莫名其妙地被公布到 Internet 上，于是变成了公开算法。因此假定加密算法公开是合理的，也是必要的。

9.4.1 纯粹密码系统

对每一个 $k\in K$，加密算法 $E_k(m)$ 是从 M 到 C 的一个一对一映射。为书写方便，记 $T_k=E_k(\cdot)$，其输入是来自 M 的消息。同样记解密算法为 $T_k^{-1}=D_k(\cdot)$。下面介绍纯粹密码系统。

定义 9.4.1 如果一个密码系统的加解密算法满足：对任意 T_i、T_j、T_k，总存在 T_s，使

$$T_iT_j^{-1}T_k = T_s \tag{9.15}$$

则该密码系统称为纯粹密码系统(pure cipher)。

尽管纯粹密码系统不要求消息空间 M 与密文空间 C 相同，但可以根据纯粹密码系统构造消息空间 M 到自身的映射，并研究这些映射的性质。

定理 9.4.1 在一个纯粹密码系统中，变换 $T_i^{-1}T_j$ 是消息空间 M 到自身的映

射。所有这些映射的全体连同映射的乘法运算，即映射的合成，构成一个 m 阶群，其中 m 是密钥个数，即从 M 到 C 的全体映射的个数。

证明：映射的合成指一个映射的像作为另一个映射的原像。首先证明形式为 $T_i^{-1}T_j$ 的 M 上的映射的集合对映射合成满足封闭性。事实上，对任意 $T_i^{-1}T_j$ 和 $T_k^{-1}T_l$，根据式(9.15)得 $T_i^{-1}T_jT_k^{-1}T_l=T_s^{-1}T_l$，因此映射运算的封闭性成立。对任意形式为 $T_i^{-1}T_j$ 的 M 上的映射，存在映射 $T_j^{-1}T_i$，使 $T_i^{-1}T_jT_j^{-1}T_i=I$，其中 I 是 M 上的恒等映射，即映射集合的单位元。这说明单位元存在，同时也说明任何形式为 $T_i^{-1}T_j$ 的 M 上的映射存在逆元素。另外容易验证结合率成立。因此定理结论成立。

为了增加安全性，自然会想到使用多个密码。把使用多个密码得到的最终结果称为这些密码的乘积。更具体地，两个密码的乘积是指一个密码加密的结果作为另一个密码的输入。对纯粹密码的乘积，有下述结论。

定理 9.4.2　两个可交换的纯粹密码的乘积仍是一个纯粹密码。

证明：假设 T 和 R 分别是两个密码的全体映射集合。因为 T 和 R 可交换，运算 TR 有意义，说明 T 的像空间正好是 R 的原像空间，RT 有意义说明 R 的像空间正好是 T 的原像空间。而 $TR=RT$ 则说明 T 和 R 都是在同一空间上的映射。因为 T 和 R 可交换，即对任意 i、j，总存在 k、l，使 $T_iR_j=R_kT_l$ 成立。因此

$$T_iR_j(T_kR_l)^{-1}T_mR_n=T_iR_jR_l^{-1}T_k^{-1}T_mR_n=R_uR_v^{-1}R_wT_rT_s^{-1}T_l$$
$$=R_hT_g=T_eR_f$$

因此 T 与 R 的乘积密码是一个纯粹密码。

图 9.2 所示是一个纯粹密码的例子。注意在所有密钥作用下，明文被分成了不同的组 M_1、M_2 和 M_3，而密文被分成了对应的组 C_1、C_2 和 C_3。从 M_i 的任意消息到 C_i 的任意密文之间有一条连线，表明存在一个密钥将该消息加密成对应的密文。而在 M_i 与 $C_j(i\neq j)$之间不存在任何连线。把这样的消息分类称为消息剩余类，而把密文的对应分类称为密文剩余类。

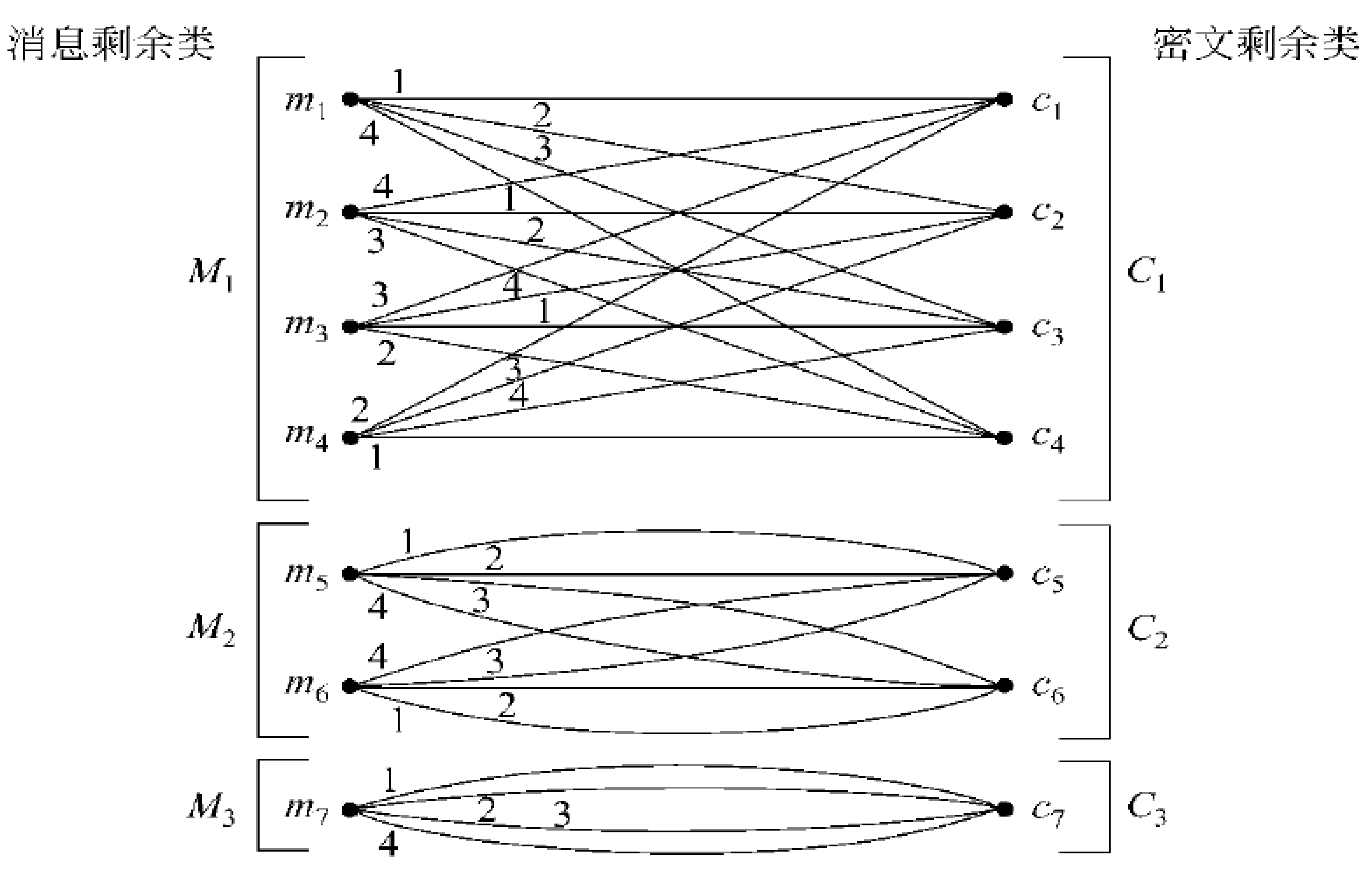


图 9.2　一个纯粹密码和对应的消息、密文剩余类

定理 9.4.3 在一个纯粹密码系统中，消息可以被分成不同的“消息剩余类”M_1，M_2，…，M_s，而密文也可以分成对应的“密文剩余类”C_1，C_2，…，C_s，且满足下述性质：

(1) 对 $i \neq j$，有 $M_i \cap M_j = \varnothing$，其中$\varnothing$为空集，且 $M = \bigcup_{i=1}^{s} M_i$ 为全部消息空间；

(2) 任意密钥对 M_i 中消息加密的结果为 C_i 中的密文；反过来，对 C_i 中任意密文的解密结果是 M_i 中的消息；

(3) 记 M_i 中的消息数为 $\varphi_i = |M_i|$，则 $|C_i| = \varphi_i$，且 φ_i 是 $|K_i|$ 的一个因子；

(4) M_i 中任意消息都可以通过 $\frac{|K|}{\varphi_i}$ 种密钥加密为 C_i 中的任意密文。对解密有类似结果。

为了描述方便，习惯上称消息自然发生的概率为先验概率，而在其他消息发生后的条件概率称为后验概率。纯粹密码的概念在于使用任何密钥加密都基本上是相同的，即没有理由说哪些密钥比其他的更好。对消息的选取也是如此。不管使用哪个密钥，也不管对哪个消息加密，加密后密文消息的概率都是相等的。为了说明这一点，注意对同一个明文消息，当使用不同密钥进行加密时，得到不同的密文，但它们属于同一个密文剩余类，不妨假设为 C_i。根据定理 9.4.3，这两个密文分别可以用 $\frac{|K|}{\varphi_i}$ 个密钥解密到 M_i 中的消息。所有密钥都等可能地使用，因此消息 $m \in M_i$ 的后验概率为

$$P(m \mid c) = \frac{P(m)P(c \mid m)}{P(c)} = \frac{P(m)P(c \mid m)}{\sum_m P(m)P(c \mid m)} = \frac{P(m)}{P(M_i)} \tag{9.16}$$

其中 $c \in C_i$。这个后验概率显然与密文 c 无关，只与 m 的先验概率及 M_i 的大小有关。类似地可以证明，密钥的后验概率也不会改变。因此得到以下定理。

定理 9.4.4 一个纯粹密码系统中消息的后验概率 $P(m|c)$ 与所选用的密钥无关；密钥的后验概率 $P(k|c)$ 的值都相同。

在实际应用中，通常密文空间与明文空间相同。比如基于 26 个英文字母表的代换密码就是明密文空间相同。容易证明，所有对 26 个英文字母的置换构成一个纯粹密码系统，其密钥个数为 $26! = 4 \times 10^{27}$。除个别小的例子外，实际使用中的密码很难满足纯粹密码系统的条件。1985 年 Kaliski Jr. 等人[3]对数据加密标准 DES 的研究发现，DES 不是一种纯粹密码。尽管实际中很少设计出纯粹密码，甚至验证一个密码系统是否为纯粹密码的过程本身就很难完成，纯粹密码以及其性质对实际密码设计还是起到重大的指导作用。

9.4.2 完备密码系统

当一个纯粹密码系统只有一个消息剩余类从而也只有一个密文剩余类时，根据式(9.16)，此时 $P(M_i)$ 为消息空间的概率，当然为 1，因此有

$$P(m \mid c) = P(m) \tag{9.17}$$

定义 9.4.2 满足等式(9.17)的密码系统称为完备密码系统。

根据定义 9.4.2，对一个完备密码系统，当观察到任何密文 c 时，它对任何明文

消息的后验概率没有任何影响。根据互信息的定义，对任意明文消息 $m\in M$ 和密文消息 $c\in C$，都有 $I(m;c)=\log_2\left(\frac{P(m|c)}{P(m)}\right)=0$，这说明任何观察到的密文 $c\in C$ 对猜测任何可能被发送的明文消息 $m\in M$ 不提供任何信息。进一步，根据式(9.12)可得到 $I(M;C)=0$，即消息空间与密文空间可以看作两个统计独立的随机变量。

定理 9.4.5　一个密码系统是完备的当且仅当对任意 $m\in M$ 和 $c\in C$，都有

$$P(c \mid m) = P(c) \tag{9.18}$$

证明：对一个完备密码系统，根据定义 9.4.2 有 $P(m|c)=P(m)$。根据概率等式 $P(m)P(c|m)=P(c)P(m|c)$，等式(9.18)与(9.17)等价，即定理得证。

对定理 9.4.5 的解释是，对一个完备密码系统，不管实际加密的是哪个明文消息，在不同密钥的作用下，任何密文都按照固有的概率(平均发生概率)发生。注意到 $P(c \mid m) = \sum_{\substack{k\in K\\ E_k(m)=c}} P(k)$，因此又可得到以下定理。

定理 9.4.6　一个密码系统是完备的当且仅当对任意 $c\in C$，$P(c \mid m) = \sum_{\substack{k\in K\\ E_k(m)=c}} P(k)$ 与消息 m 无关。

记消息的个数为 $n=|M|$，密钥的个数为 $l=|K|$，则有下述结论。

定理 9.4.7　一个完备密码系统必须满足 $l\geqslant n$。

证明：用反证法。假定 $l<n$。设 c_0 为一个有效密文，即 $P(c_0)>0$，则存在 $l_0(1\leqslant l_0\leqslant l)$ 个消息 $m\in M$，使 $m=D_k(c_0)$ 对某个密钥 $k\in K$ 成立。设 m_0 为不满足关系 $m=D_k(c_0)$ 的一个消息(共有 $n-l_0\geqslant n-l\geqslant 1$ 个这样的消息)，则有

$$P(c_0 \mid m_0) = \sum_{\substack{k\in K\\ E_k(m_0)=c_0}} P(k) = \sum_{k\in \Phi} P(k) = 0$$

但是另一方面，根据式(9.18)，一个完备密码系统应满足

$$P(c_0 \mid m_0) = P(c_0) > 0$$

此为矛盾。故定理结论得证。

定理 9.4.7 表明，一个密码系统要满足完备性，即满足密文空间与明文空间的互信息为零，所需要的密钥个数必须不少于明文的个数。但确实存在密钥个数与明文消息个数相同的情况。下面来看一个最简单的完备密码系统，所需要的密钥个数与明文消息个数相同。假定明文消息只有两个消息“是”和“非”，经过信源编码它们分别对应码子 1 与 0。所使用的密钥也有两个，即 A 与 B。使用不同密钥时的加密结果见表 9.1(表中数字为密文)。

表 9.1　加密结果

明文消息	密钥 A	密钥 B
“是”	0	1
“非”	1	0

9.4.3　密码系统的含糊度和唯一解距离

完备密码系统的特点在于密文不泄漏关于明文消息的任何信息。如果仅仅根据

获取的密文试图破译，无论破译者有多大的计算能力，都不比没有得到任何密文的情况有任何优势。但是，正如定理 9.4.7 所证明的，实现完备密码系统的必要条件是密钥的长度不少于其所加密的明文消息的长度。而在实际应用中，由于密钥的传输代价很大，通常希望使用同一密钥加密更多的消息，至少所加密的消息长度大于所使用密钥的长度。根据定义，这样的系统一定不是完备系统，因此从密文消息中一定可以获得关于明文的某些信息。但这里存在两个问题：一是在解密过程中，即使刚好能正确解密，比如经过猜测筛选甚至暴力攻击等手段测试一大批可能的密钥，测试过程也测试了加密所使用的正确密钥，但如何判断所获取的明文是正确的？这要根据破译者对明文空间的知识来解决。假如明文空间是某个长度的随机数（即完全不确定），则即使破译成功，破译者也不知道所得到的是正确的明文。如果有某种方法可以判断解密所得到的消息是否为正确的明文消息（即完全确定），则可以正确解密。但许多时候破译者对解密的结果可能有某种程度或信心的确定。另一个问题是，即使当由密文恢复出正确明文时破译者能够判断是否正确，那么由密文得到对应明文的过程可能因为需要大量计算而变得不现实，因此仍然具有实际保密效果。现代许多实际密码体制都是根据这种原理。

下面讨论第一个问题，因为第二个问题实际在现代密码学中已经解决了，即在设计密码体制时要有一个充分高的破译设计复杂度。首先来看一个简单例子：假定被加密的消息空间是标准的英文文章（由正确拼写的英文单词构成），加密所使用的方法是恺撒密码，即将字母表依次循环移位 k 个位置，这里 k 为 0～25 之间的整数。当知道这种密码后，解密所需要的计算就是测试 26 个可能的移位。假如加密所使用的密钥为 $k=3$，即字母表替换规律为 A→D，B→E，…，Z→C。则消息“creases to…”被加密成“FUHDVHV WR…”。假定加密时所使用的密钥是完全随机选取的，即 k 的值可以等可能地为 0～25 之间的任意数，则当破译者截获到密文“F”后，可以等可能地被解密为 26 个英文字母中的任一个。因此所截获的字母对判断使用了哪个密钥没有任何帮助。但是，当截获到两个字母“FU”后，因为解密后有 26 种双字母组合的情况，其中不少组合在英文中出现的概率为零，因此可以排除它们所对应的密钥，因此对密钥的判断增加了不少信息。随着所截获的字母的增加，对破译密钥的确信程度也发生变化，而且很快达到这样一种情况：其中的某一种组合成为可能，而所有其他情况都不可能，这时可以唯一确定解密密钥。图 9.3 给出了随着密文长度的增加，破译者对密钥判断的概率分布。这一概率实际为已知密文情况下密钥的条件概率，也称为后验概率，在此例中与明文消息的来源直接有关。

从图 9.3 可以看出，当截获的密文达到 5 个字母时，便可以唯一确定明文消息和密钥。把能够唯一确定明文消息的最短密文长度（上例中为字母个数）称为该密码的唯一解距离（unicity distance）。上述例子中当截获到 5 个密文字母时便可以唯一确定正确消息，但如果所加密的是其他消息，这个数字也许更小或更大。这里讨论的唯一解距离是对这一类消息源（如英文报刊文章）的最大值。当截获到的密文长度小于唯一解距离时，破译者不能唯一确定解密消息，即使具有无限的计算能力。直观上，随着所获取密文长度的增加，那个正确明文的后验概率似乎越来越大，而且也越来越

超过其他可能的明文消息的概率。但是从上述例子中就可以看出这一直觉不正确。比如当截获的密文长度为 3 时，有 3 种可能的明文消息（分别为 PERNF、LANJB 和 TIVRJ）都比正确明文的后验概率大。如果在此时就根据经验概率判断的话，很可能是错误的。

待解密文	$N=1$	$N=2$	$N=3$	$N=4$	$N=5$
F U H D V	.029	.0189			
G V I E W	.020				
H W J F X	.053	.0063			
I X K G Y	.063	.0126			
J Y L H Z	.001				
K Z M I A	.004				
L A N J B	.034	.1321	.2500		
M B O K C	.025		.0222		
N C P L D	.071	.1195			
O D Q M E	.080	.0377			
P E R N F	.020	.0818	.4389	.6327	
Q F S O G	.001				
R G T P H	.068	.0126			
S H U Q I	.061	.0881	.0056		
T I V R J	.105	.2830	.1667		
U J W S K	.025				
V K X T L	.009				
W L Y U M	.015		.0056		
X M Z V N	.002				
Y N A W O	.020				
Z O B X P	.001				
A P C Y Q	.082	.0503			
B Q D Z R	.014				
C R E A S	.028	.0377	.1111	.3673	1
D S F B T	.038	.0314			
E T G C U	.131	.0881			

图 9.3　恺撒密码密文对密钥后验概率的影响

当所截获的密文长度小于密码系统的唯一解距离时，密文对正确密钥的猜测既能提供一些信息，但又不能完全确定。当考虑某一消息空间的一般消息时，一定长度的密文对密钥的确定所提供信息的多少称为该密文长度下密钥的含糊度，记为$H(K|C)$。同样也可以定义明文的含糊度 $H(M|C)$，它们分别可以用下述公式计算：

$$H(K \mid C) = \sum_{C,K} P(C,K)\log_2 P(K \mid C) \tag{9.19}$$

$$H(M \mid C) = \sum_{C,K} P(C,M)\log_2 P(M \mid C) \tag{9.20}$$

因为含糊度实际为条件熵，因此满足条件熵的一些性质。Shannon 在其文献[2]中详细研究了含糊度的许多性质。这里引用几个结果，目的在于进一步讨论唯一解距离。

这些结果的证明可以在文献[2]中找到，因此这里不再赘述。

定理 9.4.8 对纯粹密码系统，假定其密文消息剩余类分别为 $C_1, C_2, \cdots, C_r$，则有

$$H(K \mid C) = H(K) + H(M) + \sum_{i=1}^{r} P(C_i) \log_2 \frac{P(C_i)}{|C_i|} \tag{9.21}$$

纯粹密码系统是一类具有良好密码学特性但很少在实际中用到的系统。实际中使用的是复杂的变换，在没有密钥信息的情况下，从明文到密文的变换规律很难寻找。但许多时候，这些看上去非常复杂的变换，却泄漏从明文到密文的某些统计特性，这就导致建立在统计特性上的密码分析成为可能，比如相关攻击就是这样一类使用明文、密文之间统计特性的攻击方法。

为了方便讨论唯一解距离，需要引入两个新概念。对一个信源 X，用 $H(X)$表示它的平均自信息或熵，其反映的意义是该信源的不确定程度。根据熵的计算公式(9.10)，当信源中各符号出现的概率完全相同时，信源熵达到最大，即 $H_{\max}(X) = \log_2 \frac{1}{|X|} = -\log_2 |X|$。实际信源熵与最大信源熵的比值 $h = \frac{H(X)}{H_{\max}(X)}$称为信源的相对熵，而 $R = 1 - h = 1 - \frac{H(X)}{H_{\max}(X)}$ 称为信源的冗余度。容易看出，当 $H(X) = H_{\max}(X)$，即信源达到最大不确定性时，其冗余度为 0；而当 $H(X) = 0$，即信源没有不确定性时，其冗余度为 1。在英文中，由于不同字母出现的概率不同，其信源熵达不到最大值，因此信源有冗余度。根据图 9.3 给出的单个英文字母出现的概率，可以很容易计算出英文字母表作为信源的冗余度。如果考虑英文所有单词构成的信源，其冗余度就更大，据有关数据表明，英文语言存在的冗余度在 70%～80%之间。信源的冗余度降低了信息的传输速率，但提高了对传输中所发生错误的纠错、检错能力。纠错编码的原理就是通过增加信源的冗余度来获取纠错、检错能力的。

如果一个密码系统能做的既复杂(从而很难找到简单的代数关系)又不泄漏明、密文之间的统计特性，这样的系统是比较理想的。可以将此系统理想化，提出随机密码系统的概念：一个密码系统称为随机密码系统，如果满足下述条件：

(1) 任意组合的消息都可能发生，也就是说，长度为 N 的消息个数为 $T = G^N$，其中 G 为所使用的字符个数。相应的密文消息也有 T 种可能。记 $R_0 = \log_2 G$，则 $T = 2^{R_0 N}$。通常使用二元符号进行通信，此时 $G = 2$，从而 $R_0 = 1$。

(2) 长度为 N 的消息可以分为两类：一类具有较高而且较均匀的先验概率分布；另一类总体概率也小得可以忽略不计。高概率的一类消息总数为 $S = 2^{RN}$，其中 $R = \frac{H(M)}{N}$，也就是说，R 是信源消息平均每个字符的冗余度。

(3) 解密的过程可以看作从密文到明文的一种对应关系。假定 k 个密钥被等可能地使用，则每个密文字符都对应这些密钥有 k 种对应明文的对应关系。对随机密码，假定从每个密文到 k 个明文的对应关系是随机的，即 k 个随机选取的明文。

根据式(9.11)，密钥的含糊度可表示为

$$H(K \mid C) = H(K) - I(K; C) \tag{9.22}$$

其中 $I(K;C)$是密钥与密文之间的互信息。因为在密码系统中，在给定密文的情况下，一个密钥确定唯一对应的明文(忽略计算复杂度)，因此密文给密钥所提供的信息与密文给明文提供的信息相同，即 $I(K;C)=I(M;C)$。再次根据式(9.11)得到 $I(M;C)=H(M)-H(M|C)$。我们希望知道对这样一个密码系统，当获取多少密文时，通过所有可能的解密可以唯一确定正确明文，设这个密文长度为 N，则当截获到长度为 N(比特)的密文后，明文可以唯一确定，因此条件熵 $H(M|C)=0$，于是根据随机密码的性质 2 得到 $I(K;C)=I(M;C)=H(M)=RN$，其中 R 为信源消息平均每个字符的冗余度。代入式(9.22)得 $H(K|C)=H(K)-RN$。因为消息可以唯一确定，因此密钥也可以唯一确定，即有 $H(K|C)=0$。于是得到

$$N=\frac{H(K)}{R} \tag{9.23}$$

这就是计算随机密码系统唯一解距离的公式。

实际使用的密码系统尽管不完全是随机密码，但复杂的系统可以近似看作随机密码系统。现在考虑两种极端的情况：如果明文信源的熵达到最大，即信源符号完全随机，则信源的冗余度为零，从而 $R=0$，根据式(9.23)得到 $N=\infty$，这表明无论截获到多少密文，都不能唯一确定对应的明文，因为明文是完全随机的字符。这与直观感觉是相符的。当信源的冗余度达到最大，即明文消息很确定时，这时不用截获任何密文都可以确定明文消息(这已经不是随机密码系统了)，但考虑的是由密文到明文的对应与所使用的密钥一一对应的情况，因此要确定所使用的是哪个密钥，仍然需要截获至少 $N=H(K)$长的密文。在实际密码破译中，一般需要截获到远大于唯一解距离长度的密文。

现在考虑一个实际例子：假如截获到某种已知加密算法为 DES 的密文，密文长度为 64 比特。我们知道，DES 的一组密文长度为 64 比特。假定这是一个完整的密文，而不是两个密文分别贡献一部分组成的 64 比特。利用特制的攻击工具，可以在有限时间内穷举搜索所有密钥，从而在所对应的解密中，有一个是真正的明文。但如何确定哪一个是正确明文呢？如果对信源消息完全没有预先知识，则等同于信源消息为完全随机的，根据上面的讨论知道，不能确定哪个是正确明文，即使再截获并破译无穷长度的密文都没有用。因此在实际破译中必须对信源消息有所了解。一种情况是知道信源消息为英文文字，则在解密的输出中排除那些非英文文字的，就得到部分可选的了。如果用每 8 比特表示一个英文字母(ASCII 码)，则有 8 个字母输出。在少数情况下，这仍然不能唯一确定正确的原始消息，如果有多于一个的解密输出有意义的话。这种情况下需要破译多于一组的密文。另一个特殊情况是破译者已经得到一些明密文对，这时可以唯一确定密钥，从而可以对新截获的密文正确破译。

9.5　无条件安全的实用密码体制实例分析

密码体制的理论安全性给出了很好的理论指导，但实际中很难使用像纯粹密码甚至完备密码一类的密码系统。目前实际使用的绝大多数密码体制都不具有理想的

理论安全性。如数据加密标准(DES)[4]这个对称密码,在对信源性质有所了解的情况下,给定一定密文,可以唯一确定被加密的明文,其代价是很大的计算量。目前因为单个DES的破译计算复杂度不够大而被其他密码算法取代(如AES)。公钥密码也是一样,当得到超过唯一解距离数量的密文后,就可以唯一确定明文。但由于从密文破译明文的计算量太大,而使这一过程无法实现,这是建立在计算复杂度基础上的密码体制的通病。一旦人们在计算上有所突破,比如量子计算机的实际应用,将对这类基于计算困难的密码体制带来严重的威胁。但是,在某些特殊的应用中,特别是在密钥管理方面,仍然存在无条件安全的密码方案。这些方案的特点是,即使破译者拥有无限的计算能力,仅仅根据所获取的有限的信息,无法恢复出正确的秘密信息。这里就简单介绍几个这样的实例。

9.5.1 完备门限秘密共享方案

在密码系统中,无论加密还是解密都需要密钥,而这个密钥是需要好好保存的。那么如何保存好密钥呢?最直观的方法是用人脑记忆,但这种方法的缺点是人脑记忆可能发生错误、遗忘等现象,而且随着对信息安全程度的提高,所使用的密钥越来越长,因此靠人脑记忆也越来越不实际。另一种方法是用笔将密钥记录在某个地方并保存好,但其缺点是破译者可以花代价去窃取。而且早期的密码多用于军事系统,使用密码的人员机动性较大,很难找到合适的地方存放记录的密钥。在今天的商务密码应用中也有类似的问题。假如因为任何原因导致密钥丢失,其结果是或者不能进行加密,或者不能将已加密的数据和信息解密,所造成的损失是不可估量的。如果让多个人保管同一个密钥,则被丢失的可能性会更大。

1979年,Shamir[5]提出了一种$[k,\ n]$门限方案,其思想是将要保存的秘密分发给n个成员保管,每个成员所保管的是这个秘密的一个份额。当其中任意k个成员将他们的秘密份额放到一起时,可以通过简单计算恢复出所保存的秘密。但是,任何少于k个成员的秘密份额将不足以恢复出原始秘密,即使破译者具有无限的计算能力。这样,即使其中某些成员的秘密份额丢失,只要丢失的秘密份额数量少于k,系统仍然是安全的。而如果其中某些成员忘记了自己的秘密份额,只要剩余的秘密份额数为k或更多,就能恢复出原始秘密信息。其实,Shamir秘密共享方案的设计思想非常简单:在有限域$GF(p)$上随机选取$k-1$个随机数$a_1,a_2,\cdots,a_{k-1}$,其中$a_{k-1}\neq 0$。假定要保存的秘密为a_0,则建立$k-1$次多项式$f(x)=a_0+a_1x+a_2x^2+\cdots+a_{k-1}x^{k-1}$。对每个成员,将他们的身份信息$U_i$作为输入,把输出$y_i=f(U_i)$作为成员$U_i$的秘密份额,通过秘密通道(比如离线)传给该成员。这就完成了秘密共享。为描述简单,也不失一般性,成员U_i的身份信息可以用序号i代替,于是其秘密份额为$y_i=f(i)$。为论述问题方便,下面的讨论将用这种简单形式。

现在讨论k个秘密份额信息是否能正确恢复原始秘密a_0。不失一般性,假定这k个秘密份额分别为$y_1,y_2,\cdots,y_k$,其中$y_i=f(i)$,则有

$$\begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_k \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & 2 & 2^2 & \cdots & 2^{k-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & k & k^2 & \cdots & k^{k-1} \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_{k-1} \end{pmatrix} = \boldsymbol{A}_k \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_{k-1} \end{pmatrix} \tag{9.24}$$

其中 $\boldsymbol{A}_k$ 为 k 阶范德蒙(Vandermonde)矩阵,因此是可逆的,于是得到

$$\begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_{k-1} \end{pmatrix} = \boldsymbol{A}_k^{-1} \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_k \end{pmatrix} \tag{9.25}$$

显然,给定 $y_1, y_2, \cdots, y_k$,根据式(9.25)可以唯一确定多项式 $f(x)$ 的所有系数,其中包括秘密信息 a_0。在实际计算 a_0 时,一般不用式(9.25),因为它太强大(可以计算多项式 $f(x)$ 的所有系数),但需要的只是计算 a_0。另一种更简单的方法是拉格朗日插值公式。一般地,给定 k 对(x_i, y_i),$i=1,2,\cdots,k$,可以唯一确定一个 $k-1$ 次多项式 $f(x)$,使 $y_i=f(x_i)$。事实上,$f(x)=\sum_{i=1}^{k} y_i \prod_{j\neq i} \frac{x-x_j}{x_i-x_j}$ 就是满足条件的多项式。因此当给定 $y_1, y_2, \cdots, y_k$ 后,要恢复的秘密信息为

$$a_0 = f(0) = \sum_{i=1}^{k} y_i \prod_{j\neq i} \frac{j}{j-i} \tag{9.26}$$

下面讨论当秘密份额数量少于 k 时,是否能恢复秘密信息,或能得到关于秘密信息的多少信息。由式(9.26)不难看出,无论 $y_1, y_2, \cdots, y_{k-1}$ 如何选取,因为 $\prod_{j\neq k} \frac{j}{j-k} \neq 0$,故当 y_k 取遍 $GF(p)$ 中所有可能值时,a_0 也取遍 $GF(p)$ 中所有可能值。而且当 y_k 在 $GF(p)$ 中的取值概率服从均匀分布时,a_0 在 $GF(p)$ 中的取值概率也服从均匀分布。这就证明了 $y_1, y_2, \cdots, y_{k-1}$ 与 a_0 在 $GF(p)$ 中取值的独立性,即

$$P(a_0 \mid y_1, y_2, \cdots, y_{k-1}) = P(a_0)$$

根据互信息的定义(定义 9.2.1),可以得到

定理 9.5.1 在 Shamir 秘密共享方案中,任意 $k-1$ 个秘密份额 $Y=\{y_1, y_2, \cdots, y_{k-1}\}$ 对所保护的秘密信息 $K=\{a_0\}$ 所提供的信息为零,即 $I(K;Y)=0$。

定理 9.5.1 说明 Shamir 秘密共享方案是完备的,即无论攻击者有多大的计算能力,在得到少于 k 个秘密份额的情况下对原始秘密的计算成功概率不比在没有任何秘密份额信息条件下的猜测有任何优势。如果攻击者得到更少的秘密份额,显然不会得到更多的信息。

完备秘密共享方案也称为无条件安全的,即其安全性不依赖于攻击者的计算能力。其实完备秘密共享方案的安全性更高,因为在不具备合法恢复密钥的条件下,攻击者所获得的信息对其猜测没有任何帮助。

需要说明的是:①Shamir 提出的$[k, n]$秘密共享门限方案需要有一个高级成员对秘密份额进行产生和分配;②一旦有 k 个成员将原始秘密恢复,对这些成员来说,

该秘密就不再成为秘密，因此这种方案不适合重复使用。那么，能否把门限秘密共享方案的思想应用于密码方案中，使其不再有上述两条局限性呢？答案是肯定的。最早的解决方案是 Desmedt 和 Frankel 在 1989 年提出的门限密码体制[6]，目前在这方面已经有大量研究成果。许多文章中也分析了门限方案对信息的泄漏，即信息理论意义下方案的安全性。

9.5.2 无条件安全的消息认证码

消息认证码(Message Authentication Code，MAC)是一种提供消息完整性保护的密码技术，使消息在传输过程中所发生的任何错误能被收信人检测到。目前很多通信网络都具有很好的可靠性，即数据在通过网络传输时发生错误的概率很小。再加上通信中使用纠错技术和可靠的通信协议(如 TCP)，使得实际通信中因为网络的不可靠性而发生传输错误数据的概率很小。但是，网络中存在恶意攻击，即通过网络传输的数据可能被恶意攻击者主动篡改，而这种篡改很难通过纠错码技术等手段避免。当防御系统不能工作时，检测系统就变得非常重要。消息认证码就是这样一种检测系统，即收信人对收到的信息进行消息完整性检测，如果发现消息被非法篡改，则拒绝接收(或要求重发)。因此，数据完整性保护是密码学中核心内容之一，主要用于防护网络恶意篡改攻击。

消息认证码技术使用的前提是有一个加密密钥和一个解密密钥。消息发送者对要传送的消息计算 Hash 值，用加密密钥将此 Hash 值进行加密，该加密结果连同消息一起发送给收信人。如果对消息的机密性也有需求，则将消息的 Hash 值附着在消息上，然后用加密密钥将消息及其 Hash 值一起加密。当收信人收到消息后，无论使用的是哪种方式，收信人都能使用解密密钥进行解密后得到消息和其 Hash 值，然后验证所期望的 Hash 值是否与用消息产生的 Hash 值相同。如果通信中消息被恶意篡改，则能通过消息完整性验证的可能性非常小，在实际中完全可以忽略不计。但是，无论加密算法使用的是对称密码算法还是公开密钥密码算法，一个具有无限计算能力的攻击者可以找到加密密钥，从而可以任意篡改具有消息认证码的消息，而收信人无法辨别。这类消息认证码实际是基于计算复杂性来提供安全性的。

另一类消息认证码最早是由 Gilbert 等人[7]提出来的，后来 Simmons[8]对其作了很大的扩展，使其理论更趋于成熟。这类消息认证码的安全性不依赖于攻击者的计算能力，因此它所提供的安全性是无条件的，即不假定攻击者具有有限的计算能力。下面介绍这种消息认证码及其安全性。

假定消息空间是集合 M，密钥空间是集合 K，而认证码空间是集合 C。任何一个密钥 $k \in K$ 对应于从 M 到 C 的一个单映射 $E_k: M \to C$。如果 $C=M$，则任意映射 E_k 都是集合 M 上的一个置换，这等价于许多传统的分组密码。消息认证码的安全性要求不是从截获得认证码字恢复其所对应的原始消息(这是加密体制的安全性要求)，而是攻击者能否构造一个被收信人接收的某一消息(任意消息)的合法认证码子。这比从认证码字恢复原始消息要容易得多。

攻击者可以实施下列两种攻击方法之一：

(1) 在通信开始之前，攻击者从认证码空间中选取一个认证码字 $c \in C$ 并发送给收信人。若收信人接收 c 为合法认证码子，则伪造攻击成功；如果收信人能发现 c 不是合法认证码子，则伪造攻击失败。这种攻击方法称为模仿伪造(impersonation)。

(2) 攻击者可以等待截获到通信中传输的 t 个合法认证码子 $c_1, c_2, \cdots, c_t$ 后再实施攻击。攻击者根据截获到的合法认证码字的信息和对该认证系统的知识伪造一个认证码子 c 并发送给收信人。如果收信人接收 c 为合法认证码字，并译作与 $c_1, c_2, \cdots, c_t$ 所对应的信源消息不同的消息，则称攻击成功。这种攻击称为 t 级欺骗(t-spoofing)。当 $t=1$ 时，这种攻击又称为替换攻击(substitution)。

注意在 t 级欺骗攻击中，如果攻击者构造了一个不同于 $c_1, c_2, \cdots, c_t$ 的码子 c'，并且收信人认为 c' 为合法认证码子，但其所对应的信源消息与某个已知码子 c_i 所对应的消息相同，这种情况也不算作攻击成功。为了将问题简化，在研究认证码的安全性时，通常只考虑攻击者模仿伪造成功的概率 P_I 和替换攻击的成功概率 P_S。这样，攻击者欺骗攻击成功的概率 P_d 定义为 $P_d = \max\{P_I, P_S\}$。在这样一个认证系统中，假定发信人与收信人是相互信赖的(比如他们具有共同的利益)，共同防范的是中间截获消息的攻击者。如果发信人与收信人不能相互信任，则需要其他认证码模型，如具有仲裁的消息认证码[9]。这类认证码在这里不是讨论的重点。只需要通过消息认证码提供另一类具有无条件安全的密码系统。

对于一个消息认证系统，如果攻击者的上述攻击方法都不比随机从认证码空间中选取一个码子进行模仿伪造攻击的成功概率大，则称该系统为完备认证系统，即所用的认证码为完备消息认证码。注意与完备保密概念不同的是，完备的认证系统不一定是安全的。比如当消息空间与认证码空间大小相同时，即每一个密钥对应的都是从 M 到 C 的满射时，认证码空间中的任何一个码子都代表某一信源消息，因此任取一个认证码子进行模仿欺骗，成功概率都为 1。因此，为了保证攻击者具有较低的模仿伪造成功概率，首先必须要求认证码空间 C 比消息空间 M 大得多。

对每一密钥 $k \in K$，定义合法认证码子空间为 C 的某个子集 $C_k = E_k(M) = \{c \in C: \exists m \in M \text{ s.t. } E_k(m) = c\}$。由于 E_k 为 M 到 C 的单映射，故有 $|C_k(M)| \geqslant |M|$。如果等号成立，则说明任一消息在密钥 k 作用下对应 C 中的唯一认证码子，否则说明存在消息 $m \in M$，在密钥 k 的作用下对应到不同的消息认证码子。在实际编码中，可以随机选取其所对应的认证码子，它们都是合法的。如果攻击者将其替换为表示同一消息的另一认证码子，尽管被收信人接收为合法的，因为它没有成功伪造另外的消息，因此这类攻击不算成功。当 $|C_k(M)| > |M|$ 成立时，认证码称为有分裂的。

对于一个认证码，根据所有密钥所对应的合法认证码子集合，可以构造一个 $|K| \times |C|$ 阶矩阵 $\boldsymbol{A}$，称为认证码编码矩阵，其中元素 $a_{i,j}$ 为密钥 k_i 作用下，编码后对应 c_j 的信源消息(这里假定每个集合中的元素都有一个预先的排序)。如果在密钥 k_i 作用下没有一个消息编码后对应 c_j，即 c_j 在此密钥下不是合法认证码子，则记 $a_{i,j} = 0$。根据这一原则不难看出，信源消息空间中的每个元素在编码矩阵 $\boldsymbol{A}$ 的每一行

中都至少出现一次，有些可能会出现多次（有分裂编码的情况）。例如，当 $M=\{m_1,m_2\}$，$K=\{k_1,k_2,k_3,k_4\}$，$C=\{c_1,c_2,c_3,c_4\}$时，下面的矩阵给出了一种可能的消息认证码：

$$\boldsymbol{A}=\begin{pmatrix} m_1 & m_2 & 0 & 0 \\ m_1 & 0 & m_2 & 0 \\ 0 & m_2 & 0 & m_1 \\ 0 & 0 & m_2 & m_1 \end{pmatrix} \tag{9.27}$$

对攻击者来说，由于信源消息，密钥和认证码子都是不确定的，可以把它们看作具有一定概率分布的随机变量。记 $P(c)$为 c 是合法认证码子的概率，$P(k,c)$为在密钥 k 作用下 c 是某个消息的合法认证码子的概率。定义认证函数

$$\Phi(k,c)=\begin{cases}1 & \text{当 } c \text{ 在 } k \text{ 作用下为合法认证码子时} \\ 0 & \text{其他}\end{cases} \tag{9.28}$$

则对于某一认证码子，它为合法认证码子的概率为

$$P(c\ \text{合法})=\sum_k P(k)\Phi(k,c) \tag{9.29}$$

攻击者在实施模仿伪造攻击时，一定会选取使成功概率最大的认证码子，因此有

$$P_{\mathrm{I}}=\max\{P(c)\} \tag{9.30}$$

对于模仿伪造攻击的成功概率，Simmons 在文献[9]中证明了一个重要不等式，这里作为定理列出。

定理 9.5.2 对消息认证码的模仿伪造攻击的成功概率满足

$$P_{\mathrm{I}}\geqslant 2^{-I(K;C)} \tag{9.31}$$

其中 $I(K;C)$为 K 与 C 的互信息。

证明：由式(9.30)可得

$$P_{\mathrm{I}}\geqslant \sum_c P(c)P(c\ \text{合法}) \tag{9.32}$$

式(9.32)等号成立当且仅当 $P(c$ 合法$)$对所有 c 都是相同的常数。将式(9.29)代入式(9.32)得

$$P_{\mathrm{I}}\geqslant \sum_{k,c}P(k)P(c)\Phi(k,c) \tag{9.33}$$

由于 $P(k,c)>0$ 当且仅当 $\Phi(k,c)=1$，因此式(9.33)可以等价地写为期望值的形式，即

$$P_{\mathrm{I}}\geqslant E\left[\frac{P(k)P(c)}{P(k,c)}\right]$$

对上式两端取以 2 为底的对数，得

$$\log_2 P_{\mathrm{I}}\geqslant \log_2 E\left[\frac{P(k)P(c)}{P(k,c)}\right] \tag{9.34}$$

注意对数函数是严格下凸函数，根据著名的 Jensen 不等式可得

$$\begin{aligned}\log_2 E\left[\frac{P(k)P(c)}{P(k,c)}\right] &\geqslant E\left[\log_2\frac{P(k)P(c)}{P(k,c)}\right] \\ &=H(KC)-H(K)-H(C)=-I(K;C)\end{aligned} \tag{9.35}$$

结合式(9.34)和式(9.35)得到 $\log_2 P_{\mathrm{I}} \geqslant -I(K;C)$，即式(9.31)成立，故定理得证。

注意定理 9.5.2 中使式(9.31)等号成立的充分必要条件是：

(1) $P(c$ 合法$)$对所有 c 都是相同的常数；

(2) 对所有满足 $\Phi(k,c)=1$ 的 k 和 c，$\dfrac{P(k)P(c)}{P(k,c)}$是常数。

由于模仿伪造只是攻击的一种特殊情况，因此有 $P_{\mathrm{d}} \geqslant P_{\mathrm{I}}$，于是得到认证码的信道容量定理。

定理 9.5.3(**认证信道容量定理**)

$$\log_2 P_{\mathrm{d}} \geqslant -I(K;C) \tag{9.36}$$

上述从理论上讨论了消息认证码的一些描述。下面给出几个消息认证码的实例，使读者对消息认证码的无条件安全性有更直观的认识。

考虑式(9.27)给出的认证码，即在密钥 k_1 作用下，两个信源消息 m_1 和 m_2 分别被加密为密文 c_1 和 c_2，在密钥 k_2 作用下，两个信源消息 m_1 和 m_2 分别被加密为密文 c_1 和 c_3，在密钥 k_3 作用下，两个信源消息分别被加密为密文 c_4 和 c_2，在密钥 k_4 作用下，两个信源消息分别被加密为密文 c_4 和 c_3。则当攻击者截获到密文 c_1 或 c_4 时，无论实际使用的是哪个密钥，都可以断定所发送的信源消息是 m_1，而在其他情况下可以判断信源消息为 m_2。但是，如果把编码矩阵略加修改，变为

$$\boldsymbol{A}' = \begin{pmatrix} m_1 & m_2 & 0 & 0 \\ m_2 & 0 & m_1 & 0 \\ 0 & m_1 & 0 & m_2 \\ 0 & 0 & m_2 & m_1 \end{pmatrix}$$

则容易看出，无论攻击者截获到哪个密文，除了能排除两个密钥的可能性外，对信源消息的猜测成功概率仍为$\dfrac{1}{2}$，即没有任何帮助，而且这种成功概率与攻击者的计算能力无关。如果攻击者在没有截获到任何密文前想模仿攻击，则在这两种情况下成功概率都为$\dfrac{1}{2}$。因此改造后的编码是一种无条件安全的消息认证码，或完备认证码。

上述例子或许显得有些太简单。下面给出另一个具有分裂的完备认证码。设原始信源有 4 个消息，认证码空间有 15 个码子，密钥空间有 9 个密钥，而且密钥和信源消息都等可能地使用。则下列编码矩阵所给出的是一个完备认证码，其完备性可以很容易验证。

$$\boldsymbol{A} = \left|\begin{array}{c|c|c|c} 1 & 1 & 1 & 1,2 \\ 1 & 2 & 2 & 3,4 \\ 1 & 3 & 3 & 5,6 \\ 2 & 1 & 2 & 5,6 \\ 2 & 2 & 3 & 1,2 \\ 2 & 3 & 1 & 3,4 \\ 3 & 1 & 3 & 3,4 \\ 3 & 2 & 1 & 5,6 \\ 3 & 3 & 2 & 1,2 \end{array}\right|$$

9.5.3 零知识证明的零知识性

零知识证明的概念最早是由 Goldwasser 等人[10]提出的,它已经成为密码学的基本方法之一,在密码协议中有重要应用。从字面意思看,零知识证明系统是一个交互证明过程,使得当证明完成后,验证方除了确信证明方所证明的知识外,得不到任何其他知识。也就是说,验证方在模仿证明方向第三者证明同样知识的能力并没有增加。这里的知识与信息是否一致呢?如何用信息论的方法理解零知识证明系统?这里将考虑这些问题。

简单地说,零知识证明系统是这样一个系统:有一个证明人 A,一个验证人 B,和要证明的知识 X。A 希望向 B 证明自己拥有知识 X,于是向 B 传送某些信息。B 为了确信,向 A 提问一些问题,A 对 B 所提问的问题进行答复。这种提问应答的过程可能要进行多轮,以增加 B 的确信程度。最终 B 相信 A 对 X 的掌握,但证明过程没有给 B 掌握 X 提供任何帮助。举一个简单例子:A 声称自己能计算模大素数 p 的离散对数,B 为了能确信,可以随机找一个随机数 x,计算 $y=g^x \bmod p$,其中 g 是 A 和 B 共同约定的一个数,并将 y 的值传给 A(提问)。如果 A 能够将 x 的值传给 B(应答),则 B 有理由相信 A 确实有能力计算模 p 的离散对数。但 B 可能怀疑 A 碰巧找到 x 的值而并不具有一般求离散对数的能力,于是可以进行多轮的提问。假如 A 对每一轮这样的提问都能正确应答,B 确信 A 具有计算模 p 离散对数能力的程度应该不断提高,直至几乎不怀疑。但当证明过程完成后,无论 A 与 B 之间进行几轮的提问与应答,B 都没有增加自己计算模 p 离散对数的能力。这种无零知识泄漏的过程也可以用更正规的与机器模拟不可区分的方法描述。

从上面的例子来看,零知识系统中的知识与前面提到的信息还是有区别的。比如计算模 p 的离散对数的能力就很难用确定信息来描述。证明过程没有泄漏知识,但不一定没有泄漏信息。下面从概念上(因为很难量化描述)分析什么情况下,零知识证明系统真的没有泄漏信息。

对零知识的直观解释莫过于 Jean-Jacques Quisquater 等人在 CRYPTO '89 上发表的一篇文章,它对零知识的解释大概是这样的:很久很久以前一帮强盗在山洞深处有一个密室,进入密室要有秘密口令。山洞有两个出口,中间被强盗的密室堵死了。一个偶然的机会,村民阿里巴巴偷听到了强盗的口令,想告知其他村民,但没人相信,而阿里巴巴又不想让别人知道秘密口令是什么。于是阿里巴巴想出了这样一个方法:在一个强盗外出的日子,他带领村民到山洞去验证。首先阿里巴巴自己进入山洞,其他人不知道他是从哪个洞口进去的。然后其他人任选一个洞口,让阿里巴巴从他们所在的洞口出来。阿里巴巴按照要求出来了。是否其他人就相信了呢?不完全相信,因为有人怀疑可能阿里巴巴刚好从他出来的洞口进去的呢。于是阿里巴巴进行第二次、第三次甚至更多次试验,结果都相同。后来人们相信了,但目击的人们仍然不知道进入密室的口令。有一个记者(注:当时可能没有记者这一职业和其使用的设备,但为了我们的故事,将一个现代的记者用时间飞船送到那个时代)用摄像机录制了全部过程,后来放给其他人看。可是,看过录像的人都怀疑阿里巴巴是否

真的掌握密室的口令，因为任何人都可以录制类似的录像，也就是说，真的录像(村民所选的洞口确实是随机选取的)和假的录像(村民所选的洞口是预先安排好的)不可区分。在这个故事中，阿里巴巴将其秘密告知其他村民的方法就是一个零知识证明过程。

从信息论的角度看，阿里巴巴的证明过程没泄漏任何信息，因为目击的村民除了相信阿里巴巴掌握密室口令的事实外，得不到对口令猜测的任何有用信息。同样不难看出，上面证明计算离散对数能力的证明过程也没有泄漏任何信息，因为验证人在证明过程中没有得到任何他在证明之前不知道的信息。但是，也有一些证明系统的过程会泄漏一些意图之外的信息。比如 A 要证明自己拥有某个 RSA 公钥(e,n)，即 A 掌握该公钥对应的私钥 d。B 为了验证这一点，产生一个随机数 r，并向 A 发送提问 $R=r^e \bmod n$，A 计算 $r=R^d \bmod n$ 并将 r 发送给 B，B 对比自己产生的随机数就应该相信 A 要证明的知识。同样这一过程可以进行多次。但在任何证明过程中，B 没得到任何他在证明之前不知道的信息，因此这也是一种零知识证明系统。但是，验证者 B 可以恶意地利用这一零知识证明过程获取 A 对任何消息 m 的签名 $s=m^d \bmod n$：B 在提问时向 A 发送 $R'=mr^e \bmod n$，A 在应答时将 $r'=R'^d \bmod n=rm^d \bmod n$ 传给 B，B 计算 $s=r'/r \bmod n$ 就得到所期望的签名。在这里 B 实际利用了证明系统的漏洞，从某种意义上可以称为系统的潜信道，来得到系统本身目的之外的信息。这与系统本身的零知识性并不矛盾。因此零知识证明系统是否无信息泄漏(不存在潜信道)是一个不容易确定的问题。如何确定零知识系统的信息泄漏在此留作一个公开问题。

9.6　注记

Shannon 的经典论文[1]奠基了通信的理论基础，为信源编码和信道编码提供了理论依据，也一直是这两类实际编码的追求目标(理论限)。Shannon 的经典论文[2]为信息安全的系统研究提供了重要的理论依据。但是，不同于通信的数学理论[1]对实际系统影响的是，安全系统的信息理论[2]与实际安全系统之间的关系非常微妙：许多实际系统可能根本不具有理论安全性，如公钥系统中密文对明文信息的泄漏是100%；而理论安全的系统可能根本不能用，如一次一密的加密系统。根本原因是在安全系统中用到一个极为秘密的信息，即密钥，而密钥的管理问题没有在 Shannon 的安全理论中考虑到，而在实际系统中则是非常现实的问题，因此才导致这种看似悖论的现实：理论安全的系统在实际中可能不安全，而实践证明具有实际安全性的系统可能在理论上不具有安全性。但是，也有一些理论安全与实际安全非常统一的情况，即理论安全的系统在实际中达到最好的安全性，这类系统称为完备的或完善的。这类系统多数是不涉及密钥管理问题(即假定密钥的分配与保管不存在问题，如认证码系统)，或系统本身就是密钥管理(如门限秘密共享系统)。

参 考 文 献

[1] Claude E. Shannon. A Mathematical Theory of Communication. Bell System Technical Journal, Vol. 27 (July and October 1948), pp. 379-423 and 623-656. Reprinted in D. Slepian, editor, Key Papers in the Development of Information Theory, IEEE Press, NY, 1974

[2] Claude E. Shannon. Communication theory of secrecy systems. Bell System Technical Journal, Vol. 28, 1949, pp. 656-715

[3] Burton S, Kaliski Jr. , Ronald L Rivest, Alan T. Sherman. Is DES a pure cipher?. Advances in Cryptology-CRYPTO '85: Proceedings, LNCS 218, Springer-Verlag 1986, pp. 212-226

[4] Data Encryption Standard. FIPS PUB 46, National Tech. Infor. Service, VA, 1977

[5] Shamir A. How to share a secret. Communications of the ACM, Vol. 22, 1979, pp. 612-613

[6] Desmedt Y, Frankel Y. Threshold cryptosystems. In: Advances in Cryptology-Crypto '89, Proceedings, Lecture Notes in Computer Science 435 (G. Brassard, Ed.), Springer-Verlag, 1990, pp. 307-315

[7] Gilbert E N, MacWilliams F J, Sloane N J A. Codes which detect deception. The Bell System Technical Journal, Vol. 53, No. 3, 1974, pp. 405-424

[8] Simmons G J. Message authentication without secrecy, in Secure Communications and Asymmetric Cryptosystems, G. J. Simmons, ed. , Westview Press, 1982, 105-139

[9] Simmons G J. Authentication codes that permit arbitration. Congressus Numerantium 59 (1988), 275-290

[10] Goldwasser S, Micali S, Rackoff C. The knowledge complexity of interactive proof systems, SIAM Journal on Computing, Vol. 18, No. 1, February 1989, pp. 186-208

第 10 章　频谱方法与技术

频谱方法是研究逻辑函数的重要工具，它在密码函数的特征刻画和密码分析方面显示出了独特的优越性。本章主要介绍一些在信息安全研究领域中常用的频谱方法与技术，主要内容包括布尔函数的 Walsh 谱概念及其应用；环上逻辑函数的 Chrestenson 谱概念及其基本性质；有限域上的频谱方法与技术。

10.1　Walsh 谱方法与技术

10.1.1　布尔函数的定义及其表示方法

设 n 是任一正整数，F_2 为二元域，F_2^n 表示 F_2 上的 n 维向量空间。

定义 10.1.1　设 $f(x)$是从 F_2^n 到 F_2 的映射，即对任意的 $x=(x_1,x_2,\cdots,x_n)\in F_2^n$，都有 $f(x)=f(x_1,x_2,\cdots,x_n)\in F_2$，则称 $f(x)$为 F_2^n 上的 n 元布尔函数，记为 $f:F_2^n\to F_2$ 或 $f(x),x\in F_2^n$。

为了便于研究和应用，人们在不同的情况下对布尔函数采用不同的表示。本节主要介绍布尔函数的 4 种表示形式，即真值表表示、向量表示、小项表示和多项式表示。

1. 真值表表示和向量表示

一个 n 元布尔函数 $f:F_2^n\to F_2$ 是否给定，关键在于该函数的值是否对于每一组自变量$(x_1,x_2,\cdots,x_n)$均已确定。如果把每一组自变量$(x_1,x_2,\cdots,x_n)$与其所对应的函数值全部列成表格，这种表格就叫做布尔函数 $f(x)$的**真值表**。习惯上总是按二进制表示 $x_1,x_2,\cdots,x_n$ 的值递增的顺序由上到下排列真值表（视 x_n 为最低位）。在此约定下，将表中函数值构成的长为 2^n 的行向量记为 $\boldsymbol{f}$，称为布尔函数 $f(x)$的**向量表示**。$\boldsymbol{f}$ 中的非零元素的个数称为布尔函数 $f(x)$的**重量**，记为 $W_H(f(x))$，即 $f(x)=1$ 的 x 的个数。特别地，当 $W_H(f(x))=2^{n-1}$时，则称 $f(x)$是**平衡布尔函数**。

例 10.1.1　设 $f(x)$：$F_2^2\to F_2$，其真值表如表 10.1 所示。

表 10.1　布尔函数 $f(x)$的真值表表示实例

x_1	x_2	$f(x)=f(x_1,x_2)$	x_1	x_2	$f(x)=f(x_1,x_2)$
0	0	1	1	0	1
0	1	1	1	1	0

表 10.1 所表示的函数 $f(x)$的向量表示为 $\boldsymbol{f}=(1,1,1,0)$，重量为 $W_H(f(x))=3$。

2. 小项表示

对于 $x_i,c_i\in F_2$，约定 $x_i^1=x_i,x_i^0=\bar{x}_i=1+x_i$，于是

$$x_i^{c_i}=\begin{cases}1, & x_i=c_i\\ 0, & x_i\neq c_i\end{cases}$$

设整数 $c(0\leqslant c\leqslant 2^n-1)$的二进制表示是 $c_1c_2\cdots c_n$，约定 $x^c=x_1^{c_1}x_2^{c_2}\cdots x_n^{c_n}$，它具有下述“正交性”：

$$x_1^{c_1}x_2^{c_2}\cdots x_n^{c_n}=\begin{cases}1, & (x_1,x_2,\cdots,x_n)=(c_1,c_2,\cdots,c_n)\\ 0, & (x_1,x_2,\cdots,x_n)\neq(c_1,c_2,\cdots,c_n)\end{cases}$$

由此可得到

$$f(x)=\sum_{c=0}^{2^n-1}f(c_1,c_2,\cdots,c_n)x_1^{c_1}x_2^{c_2}\cdots x_n^{c_n} \tag{10.1}$$

式(10.1)称为 $f(x)$的**小项表示**，每个被加项 $f(c_1,c_2,\cdots,c_n)x_1^{c_1}x_2^{c_2}\cdots x_n^{c_n}$ 称为一个**小项**。其中求和符号“$\sum$”是指在 F_2 上的求和。

小项表示实际上是逻辑表达方式。这种表示法常用于布尔函数的逻辑设计实现。

例 10.1.2 表 10.1 所示的布尔函数 $f(x)$：$F_2^2\rightarrow F_2$ 的小项表示为

$$\begin{aligned}f(x_1,x_2)&=1\cdot x_1^0x_2^0+1\cdot x_1^0x_2^1+1\cdot x_1^1x_2^0+0\cdot x_1^1x_2^1\\&=\bar{x}_1\bar{x}_2+\bar{x}_1x_2+x_1\bar{x}_2\end{aligned}$$

3. 多项式表示

例 10.1.2 中小项表示的 $f(x)$可以变形为 F_2 上的多项式 $f(x_1,x_2)=(x_1+1)(x_2+1)+(x_1+1)x_2+x_1(x_2+1)=1+x_1x_2$，这就得到了布尔函数 $f(x_1,x_2)$的多项式表示。一般地，将 $\bar{x}_i=1+x_i$ 代入式(10.1)，并注意到 $x_ix_i=x_i$，$x_ix_j=x_jx_i$，利用分配律并且进行同类项合并，便可使该式化为变量 $x_1,x_2,\cdots,x_n$ 的一些单项式 $x_{i_1},x_{i_2},\cdots,x_{i_r}$ 的模 2 和，即

$$f(x_1,x_2,\cdots,x_n)=a_0+\sum_{r=1}^{n}\sum_{1\leqslant i_1<i_2<\cdots<i_r\leqslant n}a_{i_1i_2\cdots i_r}x_{i_1}x_{i_2}\cdots x_{i_r} \tag{10.2}$$

$a_0,a_{i_1i_2\cdots i_r}\in F_2$，称式(10.2)为 $f(x)$**的多项式表示**。

常将式(10.2)按变元升幂及下标的字典序写出

$$\begin{aligned}f(x)=&a_0+a_1x_1+a_2x_2+\cdots+a_nx_n+a_{1,2}x_1x_2+\cdots\\&+a_{n-1,n}x_{n-1}x_na_{1,2}+\cdots+a_{1,2,\cdots,n}x_1x_2\cdots x_n\end{aligned} \tag{10.3}$$

称式(10.3)为 $f(x)$的**代数正规型**。任一确定的 n 个变元的布尔函数 $f(x)$的代数正规型式(10.3)是唯一的。一个乘积项(也称单项式)$x_{i_1}x_{i_2}\cdots x_{i_r}$ 的次数定义为 r，非零常数项的次数定义为 0，0 的次数定义为 $-\infty$。布尔函数 f 的**次数**定义为 f 的代数正规型中具有非零系数的乘积项中的最大次数，即 $\max\{k\mid a_{j_1\cdots j_k}\neq 0\}$，记为 $\deg(f)$或$\partial^0 f$。若 $\deg(f)=1$，则称 $f(x)$为**仿射布尔函数**。当 $a_0=0$ 时，仿射布尔函数被称为**线性布尔函数**；当 $\deg f\geqslant 2$ 时，称 $f(x)$为**非线性布尔函数**。

关于布尔函数 $f(x)$的下列几个事实虽然简单，但很有用。因此，写成几个引理。

引理 10.1.1 设 n 是正整数，令 $B_n=\{f(x)\mid f(x):F_2^n\rightarrow F_2\}$，即 F_2 上全体 n 元布尔函数的集合，则$|B_n|=2^{2^n}$。

证明：因为 F_2^n 中共有 2^n 个元素，对每个 $x=(x_1,x_2,\cdots,x_n)\in F_2^n$，$f(x)$的取值只有两种可能(0 或 1)，所以，$n$ 元布尔函数共有 2^{2^n} 个，即 $|B_n|=2^{2^n}$。

引理 10.1.2　设 $f(x):F_2^n\to F_2$，n 是正整数，则 $W_H(f(x))$为偶数当且仅当 $f(x)$的最高次项不出现即 $\deg f\leqslant n-1$。

证明：由式(10.3)易知，$a_{1,2,\cdots,n}=\left(\sum_{x\in F_2^n}f(x)\right)\bmod 2$，这里"$\sum$"是实数求和，则有 $a_{1,2,\cdots,n}=W_H(f(x))\bmod 2$。因此，$a_{1,2,\cdots,n}=1$ 当且仅当 $W_H(f(x))$为奇数，也等价于 $a_{1,2,\cdots,n}=0$ 当且仅当 $W_H(f(x))$为偶数当且仅当最高次项不出现即 $\deg f\leqslant n-1$。

特别地，平衡布尔函数无最高次项 $x_1x_2\cdots x_n$。

引理 10.1.3　设 n 是正整数，$f(x):F_2^n\to F_2$，$\deg f=r\ (0\leqslant r\leqslant n)$，则 $W_H(f(x))\geqslant 2^{n-r}$。

证明：通过对 n 使用数学归纳法来证明。当 $n=1$ 时，命题显然成立。假设命题对 $n-1(n\geqslant 2)$成立，下面证明命题对 n 也成立。

直接可验证，$r=0,n$ 时，有 $W_H(f(x))\geqslant 2^{n-r}$。假设 $1\leqslant r\leqslant n-1$。不难证明，任意 r 次 n 元布尔函数 $f(x_1,x_2,\cdots,x_n)$均可以分解为以下形式：

$$f=f(x_1,x_2,\cdots,x_n)=x_nf_1(x_1,x_2,\cdots,x_{n-1})+f_2(x_1,x_2,\cdots,x_{n-1}) \tag{10.4}$$

其中 $\deg f_1\leqslant r-1$，$\deg f_2\leqslant r$。

由式(10.4)可知，有 $W_H(f_2)$个形式为$(x_1,x_2,\cdots,x_{n-1},0)$的向量使得 $f=1$，有 $2^{n-1}\cdot\left(\frac{W_H(f_1)}{2^{n-1}}\cdot\frac{2^{n-1}-W_H(f_2)}{2^{n-1}}+\frac{2^{n-1}-W_H(f_1)}{2^{n-1}}\cdot\frac{W_H(f_2)}{2^{n-1}}\right)$个形式为$(x_1,x_2,\cdots,x_{n-1},1)$的向量使得 $f=1$，而且这两种形式的向量分别组成的集合的交集是空集，所以

$$\begin{aligned}W_H(f)&=W_H(f_2)+2^{m-1}\cdot\left(\frac{W_H(f_1)}{2^{n-1}}\cdot\frac{2^{n-1}-W_H(f_2)}{2^{n-1}}\right.\\&\qquad\left.+\frac{2^{n-1}-W_H(f_1)}{2^{n-1}}\cdot\frac{W_H(f_2)}{2^{n-1}}\right)\\&=W_H(f_1)+\frac{W_H(f_2)(2^{n-1}-W_H(f_1))}{2^{n-2}}\end{aligned} \tag{10.5}$$

若 $f_1\neq 0$，则由式(10.5)得 $W_H(f)\geqslant W_H(f_1)$，由归纳假设可知，$W_H(f)\geqslant W_H(f_1)\geqslant 2^{n-1-(r-1)}=2^{n-r}$。

若 $f_1=0$，则由式(10.5)得 $W_H(f)=2W_H(f_2)$，此时 $f_2\neq 0$，由归纳假设可知，$W_H(f)=2W_H(f_2)\geqslant 2\cdot 2^{n-1-r}=2^{n-r}$。

10.1.2　布尔函数的 Walsh 谱的定义及其重要性质

由于布尔函数的许多密码学性质都可以通过其 Walsh 谱予以刻画，故布尔函数的 Walsh 谱在布尔函数的性质和有关构造与应用研究中都发挥了重要的作用。

定义 10.1.2　设 $w=(w_1,w_2,\cdots,w_n)\in F_2^n$，$x=(x_1,x_2,\cdots,x_n)\in F_2^n$，$w$ 和 x 的点积(又称内积)定义为

$$w\cdot x=w_1x_1+w_2x_2+\cdots+w_nx_n\in F_2$$

n 元布尔函数 $f:F_2^n\to F_2$ 的第一种 Walsh 谱(又称线性 Walsh 谱)定义为

$$S_f(w) = \frac{1}{2^n}\sum_{x\in F_2^n} f(x)\cdot(-1)^{w\cdot x}, \quad w\in F_2^n \tag{10.6}$$

n 元布尔函数 $f:F_2^n\to F_2$ 的第二种 Walsh 谱(又称循环 Walsh 谱)定义为

$$S_{(f)}(w) = \frac{1}{2^n}\sum_{x\in F_2^n} (-1)^{f(x)+w\cdot x}, \quad w\in F_2^n \tag{10.7}$$

布尔函数及其 Walsh 谱可以相互确定。下面给出式(10.6)和式(10.7)的逆变换定理。

定理 10.1.1 任一布尔函数 $f:F_2^n\to F_2$ 与其线性 Walsh 谱 $S_f(\cdot)$和循环 Walsh 谱 $S_{(f)}(\cdot)$的关系分别为

$$f(x) = \sum_{w\in F_2^n} S_f(w)\cdot(-1)^{w\cdot x}, \quad x\in F_2^n \tag{10.8}$$

和

$$(-1)^{f(x)} = \sum_{w\in F_2^n} S_{(f)}(w)\cdot(-1)^{w\cdot x}, \quad x\in F_2^n \tag{10.9}$$

证明: 当 x、$y\in F_2^n$,$x\neq y$ 时,$\sum_{w\in F_2^n}(-1)^{w\cdot(y+x)}=0$,根据式(10.6)即知,对任一取定的 $x\in F_2^n$,都有

$$\begin{aligned}
&\sum_{w\in GF^n(2)} S_f(w)\cdot(-1)^{w\cdot x}\\
&=\sum_{w\in GF^n(2)}\left[\frac{1}{2^n}\sum_{y\in GF^n(2)} f(y)\cdot(-1)^{w\cdot y}\right]\cdot(-1)^{w\cdot x}\\
&=\frac{1}{2^n}\sum_{y\in GF^n(2)} f(y)\left[\sum_{w\in GF^n(2)}(-1)^{w\cdot(y+x)}\right]\\
&=\frac{1}{2^n}f(x)\sum_{w\in GF^n(2)}(-1)^{w\cdot(x+x)}+\frac{1}{2^n}\sum_{y\in GF^n(2),y\neq x} f(y)\sum_{w\in GF^n(2)}(-1)^{w\cdot(y+x)}\\
&=\frac{1}{2^n}f(x)\cdot 2^n+0=f(x)
\end{aligned}$$

可见式(10.8)成立。同理可证式(10.9)成立。

式(10.8)和式(10.9)也称为布尔函数的**反演公式**。

例 10.1.3 设布尔函数 $f(x_1,x_2)=x_1x_2+x_1+x_2+1$,$(x_1,x_2)\in F_2^2$,则 $f(x_1,x_2)$的函数值和线性 Walsh 谱如表 10.2 所示。

表 10.2 例 10.1.3 用表

(x_1,x_2)	$f(x_1,x_2)$	(w_1,w_2)	$S_f(w_1,w_2)$
(0,0)	0	(0,0)	$\frac{3}{4}$
(0,1)	1	(0,1)	$-\frac{1}{4}$
(1,0)	1	(1,0)	$-\frac{1}{4}$
(1,1)	1	(1,1)	$-\frac{1}{4}$

根据式(10.8)，$f(x_1,x_2)$可以表示为

$$f(x_1,x_2)=\frac{3}{4}(-1)^0-\frac{1}{4}(-1)^{x_2}-\frac{1}{4}(-1)^{x_1}-\frac{1}{4}(-1)^{x_1+x_2}$$

同样，也可以用循环 Walsh 谱表示布尔函数。

同一个布尔函数既有线性 Walsh 谱，又有循环 Walsh 谱，那么，这两类谱之间有关系吗？下面的定理回答了这个问题。

定理 10.1.2　设 $S_f(w), w\in F_2^n$ 和 $S_{(f)}(w), w\in F_2^n$ 分别是 n 元布尔函数 f：$F_2^n\to F_2$ 的线性 Walsh 谱和循环 Walsh 谱，则它们之间有以下关系：

$$S_f(w)=\begin{cases}-2S_f(w) & w\neq 0\\ 1-2S_f(w) & w=0\end{cases}\tag{10.10}$$

证明：由两种 Walsh 谱的定义并注意到$(-1)^{f(x)}=1-2f(x)$，可有

$$\begin{aligned}S_{(f)}(w)&=\frac{1}{2^n}\sum_{x\in F_2^n}(-1)^{f(x)+w\cdot x}=\frac{1}{2^n}\sum_{x\in F_2^n}[1-2f(x)]\cdot(-1)^{w\cdot x}\\&=\frac{1}{2^n}\sum_{x\in F_2^n}(-1)^{w\cdot x}-\frac{2}{2^n}\sum_{x\in F_2^n}f(x)(-1)^{w\cdot x}\\&=\begin{cases}-2S_f(w) & w\neq 0\\ 1-2S_f(w) & w=0\end{cases}\end{aligned}$$

由式(10.10)可知，只要能用其中一种谱给出布尔函数的某种密码性能的频谱特征，那么就不难给出其另一种谱的频谱特征。究竟选用哪一种谱来研究问题，则要视具体情况而定。

例 10.1.4　求布尔函数 $f(x_1,x_2)=x_1x_2+1,(x_1,x_2)\in F_2^2$ 的线性 Walsh 谱和循环 Walsh 谱。

解：首先注意到 $f(x_1,x_2)=x_1x_2+1$ 时，有

$$\begin{aligned}&x_1x_2=1\Leftrightarrow(x_1,x_2)=(1,1),\\&x_1x_2+x_1=1\Leftrightarrow(x_1,x_2)=(1,0),\\&x_1x_2+x_2=1\Leftrightarrow(x_1,x_2)=(0,1),\\&x_1x_2+x_1+x_2=1\Leftrightarrow(x_1,x_2)=(1,0)、(0,1)、(1,1),\end{aligned}$$

因而

$$\begin{aligned}S_{(f)}(0,0)&=\frac{1}{2^2}\sum_{(x_1,x_2)\in GF^2(2)}(-1)^{x_1x_2+1}=\frac{(-1)}{2^2}\sum_{(x_1,x_2)\in GF^2(2)}(-1)^{x_1x_2}\\&=\frac{(-1)}{2^2}\times(3-1)=-\frac{1}{2}\\S_{(f)}(1,0)&=\frac{1}{2^2}\sum_{(x_1,x_2)\in GF^2(2)}(-1)^{x_1x_2+1+x_1}=\frac{(-1)}{2^2}\sum_{(x_1,x_2)\in GF^2(2)}(-1)^{x_1x_2+x_1}\\&=\frac{(-1)}{2^2}\times(3-1)=-\frac{1}{2}\\S_{(f)}(0,1)&=\frac{1}{2^2}\sum_{(x_1,x_2)\in GF^2(2)}(-1)^{x_1x_2+1+x_2}=\frac{(-1)}{2^2}\sum_{(x_1,x_2)\in GF^2(2)}(-1)^{x_1x_2+x_2}\end{aligned}$$

$$=\frac{(-1)}{2^2}\times(3-1)=-\frac{1}{2}$$

$$S_{(f)}(1,1)=\frac{1}{2^2}\sum_{(x_1,x_2)\in GF^2(2)}(-1)^{x_1x_2+1+x_1+x_2}=\frac{(-1)}{2^2}\sum_{(x_1,x_2)\in GF^2(2)}(-1)^{x_1x_2+x_1+x_2}$$

$$=\frac{(-1)}{2^2}\times(1-3)=\frac{1}{2}$$

由两种谱的关系可知

$$S_f(0,0)=\frac{1}{2}[1-S_{(f)}(0,0)]=\frac{3}{4},\quad S_f(1,0)=-\frac{1}{2}S_{(f)}(1,0)=\frac{1}{4}$$

$$S_f(0,1)=-\frac{1}{2}S_{(f)}(0,1)=\frac{1}{4},\quad S_f(1,1)=-\frac{1}{2}S_{(f)}(1,1)=-\frac{1}{4}$$

下面给出的是 Walsh 谱的两个重要性质，其证明较易，可参见文献[1]、[3]，所以这里只列出而不加证明。

定理 10.1.3（Plancheral 公式） 设 $f:F_2^n\to F_2$ 为任一 n 元布尔函数，则

$$\sum_{w\in F_2^n}[S_f(w)]^2=S_f(0)=\frac{W_H(f)}{2^n}\tag{10.11}$$

此性质又称为**初值定理**。

定理 10.1.4（Parseval 公式） 设 $f:F_2^n\to F_2$ 为任一 n 元布尔函数，则

$$\sum_{w\in F_2^n}[S_{(f)}(w)]^2=1\tag{10.12}$$

此性质又称为**能量守恒定理**。

从 Walsh 谱的定义出发，容易推得以下结果。

定理 10.1.5 设 $w=(w_1,w_2,\cdots,w_n)\in F_2^n$，$x=(x_1,x_2,\cdots,x_n)\in F_2^n$，$f:F_2^n\to F_2$ 为任一 n 元布尔函数，则

(1)
$$S_f(0)=P\{f(X)=1\}=\frac{W_H(f)}{2^n}\tag{10.13}$$

当 $w\neq 0$ 时

$$S_f(w)=\frac{1}{2}-P\{f(X)=w\cdot X\}\tag{10.14}$$

(2) 对任意的 $w\in F_2^n$，有

$$S_{(f)}(w)=2P\{f(X)=w\cdot X\}-1\tag{10.15}$$

这里 $P\{\cdot\}$表示概率，$X=(X_1,X_2,\cdots,X_n)$，$X_1,X_2,\cdots,X_n$ 为某一概率空间上的 n 个相互独立且都具有均匀分布的随机变量。

定理 10.1.5 表明，Walsh 谱本质上反映了布尔函数和线性函数的符合率。定理 10.1.5又称为布尔函数 Walsh 谱的**概率表示式**，它在研究布尔函数的密码性质时发挥了重要的作用。

例 10.1.5 设布尔函数 $f(x_1,x_2)=x_1x_2+1$，$(x_1,x_2)\in F_2^2$，利用 Walsh 谱的概率表示式确定 $f(x_1,x_2)$的线性 Walsh 谱和循环 Walsh 谱。

解： 同样注意到 $f(x_1,x_2)=x_1x_2+1$ 时，有

$$x_1x_2 = 1 \Leftrightarrow (x_1, x_2) = (1,1)$$
$$x_1x_2 + x_1 = 1 \Leftrightarrow (x_1, x_2) = (1,0)$$
$$x_1x_2 + x_2 = 1 \Leftrightarrow (x_1, x_2) = (0,1)$$
$$x_1x_2 + x_1 + x_2 = 1 \Leftrightarrow (x_1, x_2) = (1,0)、(0,1)、(1,1)$$

则

$$S_{(f)}(0,0) = 2P\{f(X) = 0\} - 1 = 2P\{X_1X_2 = 1\} - 1 = 2 \times \frac{1}{4} - 1 = -\frac{1}{2}$$
$$S_{(f)}(0,1) = 2P\{f(X) = X_2\} - 1 = 2P\{X_1X_2 + X_2 = 1\} - 1 = 2 \times \frac{1}{4} - 1 = -\frac{1}{2}$$
$$S_{(f)}(1,0) = 2P\{f(X) = X_1\} - 1 = 2P\{X_1X_2 + X_1 = 1\} - 1 = 2 \times \frac{1}{4} - 1 = -\frac{1}{2}$$
$$S_{(f)}(0,0) = 2P\{f(X) = X_1 + X_2\} - 1 = 2P\{X_1X_2 + X_1 + X_2 = 1\} - 1 = 2 \times \frac{3}{4} - 1 = \frac{1}{2}$$

由两种谱的关系式或定理 10.1.5 可得线性 Walsh 谱。

10.1.3　布尔函数的 Walsh 谱的快速算法

为了叙述方便，将向量$(x_1, x_2, \cdots, x_n) \in F_2^n$用其对应的整数$x(0 \leqslant x \leqslant 2^n - 1)$来表示。设

$$\boldsymbol{f} = (f(0), f(1), \cdots, f(2^n - 1)), \quad \boldsymbol{S}_f = (S_f(0), S_f(1), \cdots, S_f(2^n - 1))$$

则

$$\boldsymbol{S}_f = 2^{-n}\boldsymbol{f}\boldsymbol{H}_n$$

其中$\boldsymbol{H}_n$由下式迭代来定义：

$$\boldsymbol{H}_0 = (1)$$
$$\boldsymbol{H}_n = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \otimes \boldsymbol{H}_{n-1} = \begin{pmatrix} \boldsymbol{H}_{n-1} & \boldsymbol{H}_{n-1} \\ \boldsymbol{H}_{n-1} & -\boldsymbol{H}_{n-1} \end{pmatrix}$$

$\otimes$表示矩阵的 Keronecker 积。因为$\boldsymbol{H}_n^2 = 2^n\boldsymbol{I}_n$（这里$\boldsymbol{I}_n$是$2^n \times 2^n$的单位矩阵）。于是

$$\boldsymbol{f} = \boldsymbol{S}_f\boldsymbol{H}_n$$

这里简要介绍一下计算布尔函数的 Walsh 谱的快速算法。设$\boldsymbol{f}^1$和$\boldsymbol{f}^2$分别表示$\boldsymbol{f}$的前一半和后一半，则

$$\boldsymbol{S}_f = 2^{-n}\boldsymbol{f}\boldsymbol{H}_n = 2^{-n}(\boldsymbol{f}^1\boldsymbol{H}_{n-1} + \boldsymbol{f}^2\boldsymbol{H}_{n-1}, \boldsymbol{f}^1\boldsymbol{H}_{n-1} - \boldsymbol{f}^2\boldsymbol{H}_{n-1})$$

按此规则一直迭代到$\boldsymbol{H}_0$就得到布尔函数的 Walsh 谱。

例 10.1.6　设布尔函数$f(x_1, x_2) = x_1x_2 + 1, (x_1, x_2) \in F_2^2$，则

$$\boldsymbol{f} = (f(0), f(1), f(2), f(3)) = (1,1,1,0)$$

故$\boldsymbol{f}^1 = (1,1), \boldsymbol{f}^2 = (1,0)$

$$H_1 = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

则

$$\boldsymbol{f}^1 \boldsymbol{H}_1 = (1,1)\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} = (2,0), \quad \boldsymbol{f}^2 \boldsymbol{H}_1 = (1,0)\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} = (1,1)$$

$$\begin{aligned} \boldsymbol{S}_f &= 2^{-2}(\boldsymbol{f}^1\boldsymbol{H}_1 + \boldsymbol{f}^2\boldsymbol{H}_1, \boldsymbol{f}^1\boldsymbol{H}_1 - \boldsymbol{f}^2\boldsymbol{H}_1) = 2^{-2}(3,1,1,-1) \\ &= \left(\frac{3}{4}, \frac{1}{4}, \frac{1}{4}, -\frac{1}{4}\right) \end{aligned}$$

10.1.4　布尔函数的自相关函数的定义及其性质

布尔函数的自相关函数可刻画布尔函数的"扩散"特征和"线性结构"特征，它与布尔函数的 Walsh 谱可以相互表出，在布尔函数的性质研究中也发挥了重要作用。

定义 10.1.3　设 $f: F_2^n \to F_2$ 是 n 元布尔函数，对 $x=(x_1, x_2, \cdots, x_n) \in F_2^n$ 和 $s=(s_1, s_2, \cdots, s_n) \in F_2^n$，记

$$x + s = (x_1 + s_1, x_2 + s_2, \cdots, x_n + s_n)$$

称

$$r_f(s) = \frac{1}{2^n} \sum_{x \in F_2^n} (-1)^{f(x+s)+f(x)} \tag{10.16}$$

为布尔函数 $f: F_2^n \to F_2$ 的自相关函数。其中 $s \in F_2^n$。

对任意两个布尔函数还可定义其互相关函数。

定义 10.1.4　设 $f_1(x)$、$f_2(x)$，$x \in F_2^n$ 是两个布尔函数，称

$$r_{f_1 f_2}(s) = \frac{1}{2^n} \sum_{x \in F_2^n} (-1)^{f_1(x+s)+f_2(x)}, \quad s \in F_2^n \tag{10.17}$$

为布尔函数 $f_1: F_2^n \to F_2$ 和 $f_2: F_2^n \to F_2$ 的互相关函数。

由 Walsh 谱的概率表达式可知，同样可以给出布尔函数自相关函数和互相关函数的概率表达式，这里不再赘述。

例 10.1.7　设布尔函数 $f(x_1, x_2) = x_1 x_2 + 1$，$(x_1, x_2) \in F_2^2$，试确定 $f(x_1, x_2)$ 的自相关函数。

解：由自相关函数的定义，则对任意的 $s=(s_1, s_2) \in F_2^2$，有

$$\begin{aligned} f(x_1 + s_1, x_2 + s_2) + f(x_1, x_2) &= (x_1 + s_1)(x_2 + s_2) + 1 + x_1 x_2 + 1 \\ &= s_2 x_1 + s_1 x_2 + s_1 s_2 \end{aligned}$$

可见当 $s=(s_1, s_2) \neq (0,0)$ 时，$f(x_1 + s_1, x_2 + s_2) + f(x_1, x_2)$ 均是平衡的，则 $r_f(s) = 0$，而 $r_f(0) = 1$。

布尔函数的相关函数与其 Walsh 谱的下述关系直至 20 世纪 90 年代初才被揭示出来，它们在考察有关特殊布尔函数的谱特征和相关函数特征时很有意义。

定理 10.1.6　设 $f: F_2^n \to F_2$ 是布尔函数，其自相关函数和 Walsh 谱分别为 $r_f(s)$，$s \in F_2^n$ 和 $S_{(f)}(w)$，$w \in F_2^n$，则

$$\frac{1}{2^n}\sum_{s\in F_2^n} r_f(s)(-1)^{w\cdot s}=[S_{(f)}(w)]^2,\quad w\in F_2^n \tag{10.18}$$

$$\sum_{w\in F_2^n}[S_{(f)}(w)]^2(-1)^{w\cdot s}=r_f(s),\quad s\in F_2^n \tag{10.19}$$

又设布尔函数 $f_1(x)$、$f_2(x)$，$x\in F_2^n$ 的互相关函数为 $r_{f_1f_2}(s)$，则有

$$\frac{1}{2^n}\sum_{s\in F_2^n} r_{f_1f_2}(s)(-1)^{w\cdot s}=S_{(f_1)}(w)\cdot S_{(f_2)}(w),\quad w\in F_2^n \tag{10.20}$$

$$\sum_{w\in F_2^n} S_{(f_1)}(w)\cdot S_{(f_2)}(w)(-1)^{w\cdot s}=r_{f_1f_2}(s),\quad s\in F_2^n \tag{10.21}$$

证明：根据式(10.16)，对任意的 $w\in F_2^n$，有

$$\begin{aligned}
&\frac{1}{2^n}\sum_{s\in F_2^n} r_f(s)\cdot(-1)^{w\cdot s}\\
&=\frac{1}{2^n}\sum_{s\in F_2^n}\left[\frac{1}{2^n}\sum_{x\in F_2^n}(-1)^{f(x+s)+f(x)}\right]\cdot(-1)^{w\cdot s}\\
&=\frac{1}{2^{2n}}\sum_{x\in F_2^n}(-1)^{f(x)}\sum_{s\in F_2^n}(-1)^{f(x+s)+w\cdot s}\\
&=\frac{1}{2^{2n}}\sum_{x\in F_2^n}(-1)^{f(x)}\sum_{y\in F_2^n}(-1)^{f(y)+w\cdot(y+x)}\\
&=\left[\frac{1}{2^n}\sum_{x\in F_2^n}(-1)^{f(x)+w\cdot x}\right]\cdot\left[\frac{1}{2^n}\sum_{y\in F_2^n}(-1)^{f(y)+w\cdot y}\right]\\
&=[S_{(f)}(w)]^2
\end{aligned}$$

即式(10.18)成立。

对任意的 $s\in F_2^n$，有

$$\begin{aligned}
&\sum_{w\in F_2^n}[S_{(f)}(w)]^2(-1)^{w\cdot s}\\
&=\sum_{w\in F_2^n}\left[\frac{1}{2^n}\sum_{v\in F_2^n} r_f(v)\ (-1)^{w\cdot v}\right]\cdot(-1)^{w\cdot s}\\
&=\frac{1}{2^n}\sum_{v\in F_2^n} r_f(v)\sum_{w\in F_2^n}(-1)^{w\cdot(v+s)}\\
&=\frac{1}{2^n}r_f(s)\sum_{w\in F_2^n}(-1)^{w\cdot(v+s)}+\frac{1}{2^n}\sum_{v\in F_2^n,v\neq s} r_f(v)\sum_{w\in F_2^n}(-1)^{w\cdot(v+s)}\\
&=r_f(s)
\end{aligned}$$

式(10.20)和式(10.21)同理可得。

例 10.1.8　设布尔函数 $f(x_1,x_2)=x_1x_2+1$，$(x_1,x_2)\in F_2^2$，验证 $f(x_1,x_2)$ 的 Walsh 谱和其自相关函数的关系式成立。

解：由例 10.1.5 和例 10.1.7 可知，

$$S_{(f)}(0,0)=-\frac{1}{2},\quad S_{(f)}(0,1)=-\frac{1}{2}$$

$$S_{(f)}(1,0)=-\frac{1}{2},\quad S_{(f)}(0,0)=\frac{1}{2}$$

$$r_f(s)=\begin{cases}0 & s\neq 0\\ 1 & s=0\end{cases}$$

$$\frac{1}{2^2}\sum_{s\in F_2^2}r_f(s)(-1)^{w\cdot s}=\frac{1}{4}\Big[\sum_{s\in F_2^2,s\neq 0}0\cdot(-1)^{w\cdot s}+r_f(0)\Big]$$

$$=\frac{1}{4}=[S_{(f)}(w)]^2$$

$$\sum_{w\in F_2^2}[S_{(f)}(w)]^2(-1)^{w\cdot s}=\frac{1}{4}\sum_{w\in F_2^2}(-1)^{w\cdot s}$$

$$=\begin{cases}1 & s=0\\ 0 & s\neq 0\end{cases}$$

10.1.5 Walsh 谱应用举例

1. 布尔函数的最佳仿射逼近

定义 10.1.5 设 $f:F_2^n\to F_2$ 是一布尔函数，若 $w^*\in F_2^n, a\in F_2$ 使

$$P\{f(X)=a+w^*\cdot X\}=\max\{P\{f(X)=b+w\cdot X\}:w\in F_2^n, b\in F_2\}$$

则称 $a+w^*\cdot x, x\in F_2^n$ 为 $f(x), x\in F_2^n$ 的最佳仿射逼近(BAA)。

由 Walsh 谱的概率表达式有

$$P\{f(X)=b+w\cdot x\}=\frac{1}{2}+\frac{(-1)^b}{2}S_{(f)}(w),\quad w\in F_2^n$$

可以给出利用 Walsh 谱求布尔函数的最佳仿射逼近的一种算法。

算法 10.1.1

第 1 步，计算 $f(x)$ 的 Walsh 谱 $S_{(f)}(w), w\in F_2^n$，找出

$$\{|S_{(f)}(w)|\,|\,w\in F_2^n\}$$

中的最大值 $|S_{(f)}(w^*)|$（可能不唯一）。

第 2 步，考虑 $S_{(f)}(w^*)$ 的正负性，若 $S_{(f)}(w^*)\geqslant 0$，则 $w^*\cdot x, x\in F_2^n$ 是 $f(x)$ 的最佳仿射逼近；否则，$w^*\cdot x+1, x\in F_2^n$ 是 $f(x)$ 的最佳仿射逼近。

例 10.1.9 求布尔函数

$$f(x_1,x_2,x_3)=x_1x_2+x_1x_3,\quad (x_1,x_2,x_3)\in F_2^3$$

的最佳仿射逼近。

解：由于

$$S_{(f)}(0,0,0)=\frac{1}{2},\quad S_{(f)}(1,0,0)=\frac{1}{2}$$

$$S_{(f)}(0,1,0)=0,\quad S_{(f)}(0,0,1)=0$$

$$S_{(f)}(1,1,0)=0,\quad S_{(f)}(1,0,1)=0$$

$$S_{(f)}(0,1,1)=\frac{1}{2},\quad S_{(f)}(1,1,1)=-\frac{1}{2}$$

故 $f(x_1,x_2,x_3)=x_1x_2+x_1x_3$ 的最佳仿射逼近是

$$0;x_1;\ x_2+x_3;\ 1+x_1+x_2+x_3,\quad (x_1,x_2,x_3)\in F_2^3$$

2. *Walsh* 谱在布尔函数相关免疫性判别中的应用

在研究基于 LFSR(线性移位寄存器)的非线性组合生成器的破译问题时,若某个驱动序列的状态 x_i 和输出的信号 $z=f(x_1,\cdots,x_i,\cdots,x_n)$ 的符合率为 $\frac{1}{2}+\varepsilon(\varepsilon>0)$,人们就称 z 和 x_i 是统计相关的。利用这种统计相关性即驱动序列的信息在输出序列中的一种泄漏(又称熵漏)实施的攻击称为**相关攻击**,且有成功的例证。为了衡量密钥流生成器抵抗相关攻击的能力,人们提出了组合函数相关免疫的概念。之后,人们对布尔函数相关免疫性的等价判别条件及相关免疫布尔函数的性质与构造等进行了诸多研究。近年来,人们还对一类特殊的相关免疫布尔函数——"饱和最优布尔函数"和用于秘密共享的"严格欺骗免疫布尔函数"进行了研究。

下面首先介绍相关免疫的定义。

定义 10.1.6　设 $f:F_2^n\to F_2$ 是任一 n 元布尔函数,而 $X_1,X_2,\cdots,X_n$ 是定义在某概率空间(Ω,F,P)上相互独立的 n 个布尔随机变量,且满足

$$P\{X_i=0\}=P\{X_i=1\}=\frac{1}{2},\quad 1\leqslant i\leqslant n$$

若对取定的正整数 $m\leqslant n$,对任意的 $1\leqslant i_1<\cdots<i_m\leqslant n$,与布尔函数 $f(x_1,x_2,\cdots,x_n)$相应的布尔随机变量 $f(X_1,X_2,\cdots,X_n)$与布尔随机向量$(X_{i_1},\cdots,X_{i_m})$都相互独立,即对任意的$(a_1,\cdots,a_m)\in F_2^m$,都有

$$P\{f(X_1,X_2,\cdots,X_n)=1,X_{i_1}=a_1,\cdots,X_{i_m}=a_m\}$$
$$=\frac{1}{2^m}P\{f(X_1,X_2,\cdots,X_n)=1\}\tag{10.22}$$

或等价地有

$$P\{f(X_1,X_2,\cdots,X_n)=1\mid X_{i_1}=a_1,\cdots,X_{i_m}=a_m\}$$
$$=P\{f(X_1,X_2,\cdots,X_n)=1\}$$

则称布尔函数 $f:F_2^n\to F_2$ 是 m 阶相关免疫的。

定理 10.1.7　布尔函数 $f(x),x\in F_2^n$ 是 m 阶相关免疫的充分必要条件是对任意的 $w\in F_2^n$: $1\leqslant W_H(w)\leqslant m$,都有

$$S_f(w)=0$$

或等价地

$$S_{(f)}(w)=0$$

证明:略。

定理 10.1.7 通常称为 **Xiao-Massey 定理**,是肖国镇教授和梅西教授于 1986 年得到的。Xiao-Massey 定理给出了相关免疫性在工程中便于验证的判别条件,它们是研究布尔函数相关免疫性时很有意义的经典结论。

例 10.1.10　3 元布尔函数

$$f(x_1,x_2,x_3,x_4)=x_1+x_1x_2+x_1x_3+x_2x_3,\quad (x_1,x_2,x_3)\in F_2^3$$

是非平衡且 1 阶相关免疫的,但不是 2 阶相关免疫的。

证明:首先根据定义来证明相关免疫性。

$$\begin{aligned}P\{f(X_1,X_2,X_3)=0\} &= P\{X_1+X_1X_2+X_1X_3+X_2X_3=0\}\\ &=P\{X_1=0\}P\{X_2X_3=0\}\\ &\quad +P\{X_1=1\}P\{(X_2+1)(X_3+1)=0\}\\ &=\frac{1}{2}\times\frac{3}{4}+\frac{1}{2}\times\frac{3}{4}=\frac{3}{4}\end{aligned}$$

$$P\{f(X_1,X_2,X_3)=X_1\}=P\{X_1X_2+X_1X_3+X_2X_3=0\}=\frac{1}{2}$$

$$\begin{aligned}P\{f(X_1,X_2,X_3)=X_2\} &= P\{X_1+X_1X_2+X_1X_3+X_2X_3=X_2\}\\ &=P\{X_1=0\}P\{X_2(X_3+1)=0\}\\ &\quad +P\{X_1=1\}P\{X_3(X_2+1)=1\}\\ &=\frac{1}{2}\times\frac{3}{4}+\frac{1}{2}\times\frac{1}{4}=\frac{1}{2}\end{aligned}$$

$$\begin{aligned}P\{f(X_1,X_2,X_3)=X_3\} &= P\{X_1+X_1X_2+X_1X_3+X_2X_3=X_3\}\\ &=P\{X_1=0\}P\{X_3(X_2+1)=0\}\\ &\quad +P\{X_1=1\}P\{X_2(X_3+1)=1\}\\ &=\frac{1}{2}\times\frac{3}{4}+\frac{1}{2}\times\frac{1}{4}=\frac{1}{2}\end{aligned}$$

可见 $f(X_1,X_2,X_3)=X_1+X_1X_2+X_1X_3+X_2X_3$ 分布不均匀但与 X_1、X_2、X_3 都相互独立,即布尔函数 $f(x_1,x_2,x_3,x_4)=x_1+x_1x_2+x_1x_3+x_2x_3$ 是非平衡且 1 阶相关免疫的。

又因为

$$\begin{aligned}P\{f(X_1,X_2,X_3)=X_1+X_2\} &= P\{X_1+X_1X_2+X_1X_3+X_2X_3=X_1+X_2\}\\ &=P\{X_1=0\}P\{X_2(X_3+1)=0\}\\ &\quad +P\{X_1=1\}P\{X_3(X_2+1)=0\}\\ &=\frac{1}{2}\times\frac{3}{4}+\frac{1}{2}\times\frac{3}{4}=\frac{3}{4}\end{aligned}$$

因而 $f(X_1,X_2,X_3)=X_1+X_1X_2+X_1X_3+X_2X_3$ 与 X_1+X_2 不相互独立,故布尔函数

$$f(x_1,x_2,x_3,x_4)=x_1+x_1x_2+x_1x_3+x_2x_3,\quad (x_1,x_2,x_3)\in F_2^3$$

不可能是 2 阶相关免疫的。

下面利用谱判别条件证明:由上面的证明过程及 Walsh 谱的概率表达式可得布尔函数

$$f(x_1,x_2,x_3,x_4)=x_1+x_1x_2+x_1x_3+x_2x_3,\quad (x_1,x_2,x_3)\in F_2^3$$

的 Walsh 循环谱满足:

$$S_{(f)}(0,0,0)=\frac{1}{2},\quad S_{(f)}(1,0,0)=S_{(f)}(0,1,0)=S_{(f)}(0,0,1)=0$$

$$S_{(f)}(1,1,0)=\frac{1}{2},\quad S_{(f)}(1,0,1)=\frac{1}{2}$$

$$S_{(f)}(0,1,1)=-\frac{1}{2},\quad S_{(f)}(1,1,1)=0$$

当 $W_H(w)=1$，即 $S_{(f)}(1,0,0)=S_{(f)}(0,1,0)=S_{(f)}(0,0,1)=0$，而 $S_{(f)}(1,1,0)=\frac{1}{2}\neq 0$ 且 $W_H(1,1,0)=2$，由 Xiao-Massey 定理可知，该函数是非平衡且 1 阶相关免疫的，但不是 2 阶相关免疫的。

3. *Walsh 谱在扩散特性研究中的应用*

为了研究布尔函数的扩散特性，人们提出了扩散准则的概念。

定义 10.1.7 设 $f: F_2^n \to F_2$ 是一个 n 元布尔函数，若对所有汉明重量为 1 的 $s\in F_2^n$ 布尔函数

$$f(x+s)+f(x),\quad x\in F_2^n$$

都是平衡的，即 $f(\cdot)$ 的自相关函数 $r_f(\cdot)$ 满足

$$r_f(s)=0,\quad s\in F_2^n,\quad W_H(s)=1 \tag{10.23}$$

则称布尔函数 $f(x)$，$x\in F_2^n$ 是满足严格雪崩准则的。

定义 10.1.8 设 $f: F_2^n\to F_2$ 是一个 n 元布尔函数。

(1) 对于 $s\in F_2^n$，若布尔函数 $f(x+s)+f(x)$，$x\in F_2^n$ 是平衡的，即 $f(\cdot)$ 的自相关函数 $r_f(\cdot)$ 在 s 处满足 $r_f(s)=0$，则称 $f(x)$，$x\in F_2^n$ 关于 $s\in F_2^n$ 是满足扩散准则的。

(2) 若对所有的 $s\in F_2^n$，$1\leqslant W_H(s)\leqslant k$，布尔函数 $f(x+s)+f(x)$，$x\in F_2^n$ 都是平衡的，即 $f(\cdot)$ 的自相关函数 $r_f(\cdot)$ 满足

$$r_f(s)=0,\quad s\in F_2^n,\quad 1\leqslant W_H(s)\leqslant k \tag{10.24}$$

则称布尔函数 $f(x)$，$x\in F_2^n$ 是满足 k 次扩散准则的。

由定理 10.1.6 易得以下定理。

定理 10.1.8 (1) 布尔函数 $f(x)$，$x\in F_2^n$ 满足严格雪崩准则的充分必要条件是对所有的 $s^{(i)}\in F_2^n$，$W_H(s^{(i)})=1$ 且 $s^{(i)}$ 的第 i 个分量为 1，$1\leqslant i\leqslant n$，都有

$$\sum_{w\in F_2^n} S_{(f)}^2(w)\cdot(-1)^{w_i}=0 \tag{10.25}$$

(2) 布尔函数 $f(x)$，$x\in F_2^n$ 关于 $s\in F_2^n$ 满足扩散准则的充分必要条件是

$$\sum_{w\in F_2^n} S_{(f)}^2(w)\cdot(-1)^{w\cdot s}=0 \tag{10.26}$$

因而，布尔函数 $f(x)$，$x\in F_2^n$ 满足 k 次扩散准则的充分必要条件是对所有的 $s\in F_2^n$，$1\leqslant W_H(s)\leqslant k$，都有

$$\sum_{w\in F_2^n} S_{(f)}^2(w)\cdot(-1)^{w\cdot s}=0 \tag{10.27}$$

例 10.1.11 设布尔函数 $f(x_1,x_2)=x_1x_2+1$，$(x_1,x_2)\in F_2^2$，由例 10.1.8 可知，

$$S_{(f)}(0,0)=-\frac{1}{2},\quad S_{(f)}(0,1)=-\frac{1}{2},\quad S_{(f)}(1,0)=-\frac{1}{2},\quad S_{(f)}(0,0)=\frac{1}{2}$$

再由定理 10.1.8 可知，$f(x_1,x_2)=x_1x_2+1$ 满足 2 次扩散准则。事实上，

$$r_f(s)=\begin{cases}0 & s\neq 0\\ 1 & s=0\end{cases}$$

可知 $f(x_1,x_2)=x_1x_2+1$ 是满足 2 次扩散准则。

10.2 Chrestenson 谱方法与技术

10.2.1 m 值逻辑函数的定义

以下假定 $m\geqslant 2$ 是任一取定的正整数，整数模 m 的剩余类环记为 Z_m。又对任一取定的正整数 n，以 Z_m^n 表示 n 个 Z_m 的笛卡儿积。

定义 10.2.1 设 $f(x)$ 是从 Z_m^n 到 Z_m 的一个映射，即对于任一 $x=(x_1,x_2,\cdots,x_n)\in Z_m^n$，都有 $f(x)=f(x_1,x_2,\cdots,x_n)\in Z_m$，则称 $f(x)$ 为 Z_m^n 上的 n 元 m 值逻辑函数，记为 $f:Z_m^n\rightarrow Z_m$ 或 $f(x),x\in Z_m^n$。

易知，Z_m^n 上的 n 元 m 值逻辑函数共有 m^{m^n} 个。

当 m 是素数时，m 值逻辑函数如同布尔函数一样也有真值表表示、小项表示和多项式表示等表示方法。

例 10.2.1 $m=3$ 时，若 1 元 3 值逻辑函数是

$$f(0)=2,\quad f(1)=1,\quad f(2)=0$$

则有

$$f(x)=(x+1)(x+2)+2x(x+1)=2x+2,\quad x\in Z_3$$

当 m 是合数时，m 值逻辑函数可有真值表表示和类似于小项表示的如下表示：

$$f(x)=\sum_{c\in Z_m^n} f_1(c)I_{\{c\}}(x) \tag{10.28}$$

其中

$$I_{\{c\}}(x)=\begin{cases}1 & x=c\\ 0 & x\neq c\end{cases}$$

值得一提的是，尽管此时 Z_m 上的任一多项式都可以表示为一个 m 值逻辑函数，但是任一 m 值逻辑函数却不一定有多项式表示，且有多项式表示时也不一定唯一。

10.2.2 Chrestenson 谱的定义及其基本性质

如同布尔函数的许多密码学性质都可以通过其 Walsh 谱予以刻画一样，m 值逻辑函数的许多密码学性质也都可以通过其 Chrestenson 谱予以刻画，故 m 值逻辑函数的 Chrestenson 谱在 m 值逻辑函数的性质研究中仍能发挥重要作用。

定义 10.2.2 设 $x=(x_1,x_2,\cdots,x_n)\in Z_m^n$，$w=(w_1,w_2,\cdots,w_n)\in Z_m^n$，$x$ 和 w 的点积定义为

$$w\cdot x=w_1x_1+w_2x_2+\cdots+w_nx_n\in Z_m$$

设 $f(x),x\in Z_m^n$ 为 n 元 m 值逻辑函数，称

$$S_f(w)=\frac{1}{m^n}\sum_{x\in Z_m^n} f(x)\cdot u^{-w\cdot x},\quad w\in Z_m^n \tag{10.29}$$

和

$$S_{(f)}(w)=\frac{1}{m^n}\sum_{x\in Z_m^n}u^{f(x)-w\cdot x},\quad w\in Z_m^n \tag{10.30}$$

分别为 $f(x)$ 的第一种 Chrestenson 谱(又称线性 Chrestenson 谱)和第二种 Chrestenson 谱(又称循环 Chrestenson 谱)。其中 $u=\exp\left(\frac{2\pi i}{m}\right)$, $i=\sqrt{-1}$。

显然,当 $m=2$ 时,Chrestenson 谱就是 Walsh 谱,这说明 Chrestenson 谱是 Walsh 谱的一种推广。

下面介绍式(10.29)和式(10.30)的反演公式。

定理 10.2.1　任一 n 元 m 值逻辑函数 $f(x)$ 与其线性 Chrestenson 谱和循环 Chrestenson 谱的关系分别为

$$f(x)=\sum_{w\in Z_m^n}S_f(w)\cdot u^{w\cdot x},\quad x\in Z_m^n \tag{10.31}$$

和

$$u^{f(x)}=\sum_{w\in Z_m^n}S_{(f)}(w)\cdot u^{w\cdot x},\quad x\in Z_m^n \tag{10.32}$$

证明:由 Chrestenson 谱的定义易得。

例 10.2.2　设 2 元 3 值逻辑函数

$$f(x_1,x_2)=x_1x_2,\quad (x_1,x_2)\in Z_3^2$$

试确定其循环 Chrestenson 谱。

解:由 Chrestenson 谱的定义有

$$S_{(f)}(0,0)=\frac{1}{3^2}\sum_{(x_1,x_2)\in Z_3^2}u^{x_1x_2}=\frac{1}{9}(5u^0+2u^1+2u^2)=\frac{1}{3}$$

$$\begin{aligned}S_{(f)}(w_1,w_2)&=\frac{1}{3^2}\sum_{(x_1,x_2)\in Z_3^2}u^{x_1x_2-w_1x_1-w_2x_2}\\&=\frac{1}{3^2}\sum_{(x_1,x_2)\in Z_3^2}u^{(x-w_2)(x_2-w_1)-w_1w_2}=\frac{u^{-w_1w_2}}{3^2}\sum_{(y_1,y_2)\in Z_3^2}u^{y_1y_2}\\&=u^{-w_1w_2}S_{(f)}(0,0)=\frac{u^{-w_1w_2}}{3},\quad (w_1,w_2)\in Z_3^2\end{aligned}$$

即

$$S_{(f)}(0,0)=S_{(f)}(0,1)=S_{(f)}(1,0)=S_{(f)}(0,2)=S_{(f)}(2,0)=\frac{1}{3}$$

$$S_{(f)}(1,1)=\frac{1}{3}u^2,\quad S_{(f)}(1,2)=S_{(f)}(2,1)=\frac{1}{3}u,\quad S_{(f)}(2,2)=\frac{1}{3}u^2$$

例 10.2.3　设 2 元 4 值逻辑函数

$$f(x_1,x_2)=x_1x_2,\quad (x_1,x_2)\in Z_4^2$$

试确定其循环 Chrestenson 谱。

解:由 Chrestenson 谱的定义有

$$S_{(f)}(0,0)=\frac{1}{4^2}\sum_{(x_1,x_2)\in Z_4^2} i^{x_1x_2}=\frac{1}{16}(8i^0+2i^1+4i^2+2i^3)=\frac{1}{4}$$

于是有

$$S_{(f)}(w_1,w_2)=\frac{1}{4^2}\sum_{(x_2,x_2)\in Z_4^2} i^{x_1x_2-w_1x_1-w_2x_2}=\frac{1}{4^2}\sum_{(x_1,x_2)\in Z_4^2} i^{(x_1-w_2)(x_2-w_1)+w_1w_2}$$

$$=\frac{i^{w_1w_2}}{4^2}\sum_{(y_1,y_2)\in Z_4^2} i^{y_1y_2}=i^{w_1w_2}S_{(f)}(0,0)=\frac{i^{w_1w_2}}{4},\quad (w_1,w_2)\in Z_4^2$$

即

$$S_{(f)}(0,0)=S_{(f)}(1,0)=S_{(f)}(0,1)=S_{(f)}(2,0)=S_{(f)}(0,2)$$

$$=S_{(f)}(3,0)=S_{(f)}(0,3)=S_{(f)}(2,2)=\frac{1}{4}$$

$$S_{(f)}(1,1)=-\frac{1}{4}i;\quad S_{(f)}(2,1)=S_{(f)}(1,2)=-\frac{1}{4}$$

$$S_{(f)}(3,1)=S_{(f)}(1,3)=\frac{1}{4}i$$

$$S_{(f)}(3,2)=S_{(f)}(2,3)=-\frac{1}{4};\quad S_{(f)}(3,3)=-\frac{1}{4}i$$

由 Chrestenson 谱的定义可推出关于 Chrestenson 谱的下面几个基本性质。

定理 10.2.2 设 $f(x),g(x),x\in Z_m^n$ 都是 n 元 m 值逻辑函数，则 Chrestenson 谱有以下性质：

(1)
$$S_{fg}(w)=\sum_{v\in Z_m^n} S_f(v)S_g(w-v),\quad w\in Z_m^n \tag{10.33}$$

(2)
$$S_{(f+g)}(w)=\sum_{v\in Z_m^n} S_{(f)}(v)S_{(g)}(w-v),\quad w\in Z_m^n \tag{10.34}$$

定理 10.2.3 设 m 值逻辑函数 $f(x),x\in Z_m^n$ 的线性 Chrestenson 谱为 $S_f(w)$，$w\in Z_m^n$，则

$$\sum_{w\in Z_m^n} |S_f(w)|^2=S_{f^2}(\mathbf{0}) \tag{10.35}$$

此性质又称为**初值定理**。这里 $|S_f(w)|$ 表示 $S_f(w)$ 的模即绝对值。

定理 10.2.4 设 m 值逻辑函数 $f(x),x\in Z_m^n$ 的循环 Chrestenson 谱为 $S_{(f)}(w)$，$w\in Z_m^n$，则

$$\sum_{w\in Z_m^n} |S_{(f)}(w)|^2=1 \tag{10.36}$$

此性质又称为**能量守恒定理**。这里 $|S_{(f)}(w)|$ 表示 $S_{(f)}(w)$ 的模即绝对值。

从 Chrestenson 谱的定义出发，容易推得以下定理

定理 10.2.5 设 $f(x_1,x_2,\cdots,x_n),(x_1,x_2,\cdots,x_n)\in Z_m^n$ 是任一 m 值逻辑函数，对任意的 $w=(w_1,w_2,\cdots,w_n)\in Z_m^n$，以 $w^*=(w_1^*,w_2^*,\cdots,w_n^*)$ 表示它在加法群 Z_m^n 中的逆元，记

$$X=(X_1,X_2,\cdots,X_n),\quad w^*\cdot X=w_1^*X_1+w_2^*X_2+\cdots+w_n^*X_n$$

则 m 值逻辑函数 $f(\cdot)$ 的线性和循环 Chrestenson 谱可分别表示为

$$S_f(w) = \sum_{i=0}^{m-1}\sum_{j=0}^{m-1} iu^j P\{f(X) = i, w^* \cdot X = j\}, \quad w \in Z_m^n \tag{10.37}$$

$$\begin{aligned}S_{(f)}(w) &= \sum_{r=0}^{m-1} u^r P\{f(X) - w \cdot X = r\}\\ &= \sum_{r=0}^{m-1} u^r P\{f(X) = w \cdot X + r\}\\ &= \sum_{r=0}^{m-1} u^r P\{f(x) + w^* \cdot X = r\}\\ &= \sum_{i=0}^{m-1}\sum_{j=0}^{m-1} u^{i+j} P\{f(X) = i, w^* \cdot X = j\}, \quad w \in Z_m^n\end{aligned} \tag{10.38}$$

式(10.37)和式(10.38)也称为 m 值逻辑函数的 Chrestenson 谱的**概率表示式**。

例 10.2.4　设 2 元 4 值逻辑函数

$$f(x_1, x_2) = x_1 x_2, \quad (x_1, x_2) \in Z_4^2$$

利用 Chrestenson 谱的概率表示式来确定 Chrestenson 谱。

解：首先注意到

$$x_1x_2 = 0 \Leftrightarrow (x_1, x_2) = (0,0),(0,1),(0,2),(0,3),(1,0),(2,0),(3,0),(2,2)$$
$$x_1x_2 = 1 \Leftrightarrow (x_1, x_2) = (1,1),(3,3)$$
$$x_1x_2 = 2 \Leftrightarrow (x_1, x_2) = (1,2),(2,1),(2,3),(3,2)$$
$$x_1x_2 = 3 \Leftrightarrow (x_1, x_2) = (1,3),(3,1)$$

当 $(w_1^*, w_2^*) = (0,0)$ 时

$$P\{X_1X_2 = j, 0 \cdot X_1 + 0 \cdot X_2 = 0\} = P\{X_1X_2 = j\}$$

即

$$P\{X_1X_2 = 0, 0 \cdot X_1 + 0 \cdot X_2 = 0\} = \frac{1}{2}$$
$$P\{X_1X_2 = 2, 0 \cdot X_1 + 0 \cdot X_2 = 0\} = \frac{1}{4}$$
$$P\{X_1X_2 = 1, 0 \cdot X_1 + 0 \cdot X_2 = 0\}$$
$$= P\{X_1X_2 = 3, 0 \cdot X_1 + 0 \cdot X_2 = 0\} = \frac{1}{8}$$
$$P\{X_1X_2 = j, 0 \cdot X_1 + 0 \cdot X_2 = k\} = 0, \quad k \neq 0$$

则

$$\begin{aligned}S_{(f)}(0,0) &= \sum_{j=0}^{3}\sum_{k=0}^{3} i^{j+k} P\{X_1X_2 = j, 0 \cdot X_1 + 0 \cdot X_2 = k\}\\ &= \frac{1}{2} \times i^0 + \frac{1}{8} \times i^1 + \frac{1}{4} \times i^2 + \frac{1}{8} \times i^3 = \frac{1}{4}\end{aligned}$$

当 $(w_1^*, w_2^*) = (1,1)$ 时

$$P\{X_1X_2 = 0, X_1 + X_2 = 0\} = P\{X_1X_2 = 3, X_1 + X_2 = 0\} = \frac{1}{8}$$
$$P\{X_1X_2 = 1, X_1 + X_2 = 0\} = P\{X_1X_2 = 2, X_1 + X_2 = 0\} = 0$$

$$P\{X_1X_2=0,X_1+X_2=1\}=P\{X_1X_2=2,X_1+X_2=1\}=\frac{1}{8}$$

$$P\{X_1X_2=1,X_1+X_2=1\}=P\{X_1X_2=3,X_1+X_2=1\}=0$$

$$P\{X_1X_2=0,X_1+X_2=2\}=P\{X_1X_2=1,X_1+X_2=2\}=\frac{1}{8}$$

$$P\{X_1X_2=2,X_1+X_2=2\}=P\{X_1X_2=3,X_1+X_2=2\}=0$$

$$P\{X_1X_2=0,X_1+X_2=3\}=P\{X_1X_2=2,X_1+X_2=3\}=\frac{1}{8}$$

$$P\{X_1X_2=1,X_1+X_2=3\}=P\{X_1X_2=3,X_1+X_2=3\}=0$$

则

$$\begin{aligned}S_{(f)}(3,3)&=\sum_{j=0}^{3}\sum_{k=0}^{3}i^{j+k}P\{X_1X_2=j,X_1+X_2=k\}\\&=\frac{1}{8}\times i^0+\frac{1}{8}\times i^3+\frac{1}{8}\times i^1+\frac{1}{8}\times i^3+\frac{1}{8}\\&\quad\times i^2+\frac{1}{8}\times i^3+\frac{1}{8}\times i^3+\frac{1}{8}\times i^5\\&=-\frac{1}{4}i\end{aligned}$$

其余情况类似可得。

10.2.3 两种 Chrestenson 谱之间的关系

首先给出一个重要引理。

引理 10.2.1 设 $X=(X_1,X_2,\cdots,X_n)$是概率空间$(\Omega,\mathcal{F},P)$上的 n 维 m 值随机变量，则 $X_1,X_2,\cdots,X_n$ 相互独立且都具有均匀分布的充分必要条件是对任意的 $w\in Z_m^n$ 且 $w\neq0$，有

$$P\{w\cdot X=r\}=\begin{cases}\dfrac{d}{m}, & r=0,d,\cdots,(m_1-1)d\\0, & r\neq0,d,\cdots,(m_1-1)d\end{cases},\quad r\in Z_m \tag{10.39}$$

其中 d 为 $w=(w_1,w_2,\cdots,w_n)$中的 $w_1,w_2,\cdots,w_n$ 和 m 这 $n+1$ 个数的最大公约数，而 $m_1=\dfrac{m}{d}$。

定理 10.2.6 设 $f(x),x\in Z_m^n$ 是任一 m 值逻辑函数，对任一取定的 $w\in Z_m^n$，记

$$y_i=\sum_{j=0}^{m-1}u^jP\{f(X)=i,w^*\cdot X=j\},\quad i=0,1,\cdots,m-1$$

则有

$$\sum_{i=0}^{m-1}y_i=I_{\{0\}}(w)=\begin{cases}1 & w=0\\0 & w\neq0\end{cases} \tag{10.40}$$

证明：当 $w=0$ 时，有 $w^*=0$，注意到

$$P\{f(X)=i,w^*\cdot X=j\}=\begin{cases}0 & j\neq0\\P\{f(x)=i\} & j=0\end{cases}$$

即知

$$\sum_{i=0}^{m-1} y_i = \sum_{i=0}^{m-1}\sum_{j=0}^{m-1} u^j P\{f(X)=i, w^* \cdot X = j\}$$
$$= \sum_{i=0}^{m-1} P\{f(X)=i\} = 1$$

当 $w \neq 0$ 时，有 $w^* \neq 0$，记 $w^*=(w_1^*, w_2^*, \cdots, w_n^*)$，又记 $w_1^*, w_2^*, \cdots, w_n^*$ 和 m 的最大公约数为 d，而 $m_1 = \frac{m}{d}$，注意到 $u^m - 1 = 0$，由式(10.39)即知

$$\sum_{i=0}^{m-1} y_i = \sum_{i=0}^{m-1}\sum_{j=0}^{m-1} u^j P\{f(X)=i, w^* \cdot X = j\}$$
$$= \sum_{j=0}^{m-1} u^j \left[\sum_{i=0}^{m-1} P\{f(x)=i, w^* \cdot X = j\}\right]$$
$$= \sum_{j=0}^{m-1} u^j P\{w^* \cdot X = j\} = \sum_{k=0}^{m_1-1} u^{kd} \cdot \frac{d}{m}$$
$$= \frac{d}{m} \cdot \frac{u^{m_1 d} - 1}{u^d - 1} = \frac{d}{m} \cdot \frac{u^m - 1}{u^d - 1}$$
$$= 0$$

综上所述，可知式(10.40)成立。

对 $\lambda = 0, 1, \cdots, m-1$，以下记 m 值逻辑函数 $f(x)+\lambda, x \in Z_m^n$ 的线性 Chrestenson 谱为 $S_{f+\lambda}(w), w \in Z_m^n$；又对 $k=1,2,\cdots,m-1$，记 m 值逻辑函数 $kf(x), x \in Z_m^n$ 的循环 Chrestenson 谱为 $S_{(kf)}(w), w \in Z_m^n$。

下面的定理首先给出用 m 值逻辑函数

$$f(x)+\lambda, \quad x \in Z_m^n, \quad \lambda = 0, 1, \cdots, m-1$$

的线性 Chrestenson 谱 $S_{f+\lambda}(w), w \in Z_m^n$ 线性表出 m 值逻辑函数

$$kf(x), \quad x \in Z_m^n, \quad k = 1, 2, \cdots, m-1$$

的循环 Chrestenson 谱 $S_{(kf)}(w), w \in Z_m^n$ 的关系式。

定理 10.2.7　对任意的 n 元 m 值逻辑函数 $f(x), x \in Z_m^n$ 及任意的 $w \in Z_m^n$ 和 $k=1,2,\cdots,m-1$，都有

$$S_{(kf)}(w) = \frac{1}{m}\sum_{i=0}^{m-1} u^{ki}[S_{f+m-1-i}(w) - S_{f+m-i}(w)] \tag{10.41}$$
$$= \frac{1}{m}\sum_{\lambda=0}^{m-1}[u^{k(m-\lambda-1)} - u^{k(m-\lambda)}]S_{f+\lambda}(w)$$
$$= \frac{u^{-k}-1}{m}\sum_{\lambda=0}^{m-1} u^{-k\lambda} S_{f+\lambda}(w) \tag{10.42}$$

证明： $y_i, i=0,1,\cdots,m-1$ 的定义如定理 10.2.6，则

$$S_{f+\lambda}(w) = \sum_{i=0}^{m-1}(i+\lambda)_{(\mathrm{mod}\ m)}\left[\sum_{j=0}^{m-1} u^j P\{f(X)=i, w^* \cdot X = j\}\right]$$
$$= \sum_{i=0}^{m-1}(i+\lambda)_{(\mathrm{mod}\ m)} y_i, \quad \lambda = 0, 1, \cdots, m-1 \tag{10.43}$$

则

$$S_{(kf)}(w) = \sum_{i=0}^{m-1} u^{ki} y_i, \quad k = 1,2,\cdots,m-1 \tag{10.44}$$

解下列组成的方程组

$$\begin{cases} y_0 + y_1 + y_2 + \cdots + y_{m-2} + y_{m-1} = \boldsymbol{I}_{\{0\}}(w) \\ 0 + y_1 + 2y_2 + \cdots + (m-2)y_{m-2} + (m-1)y_{m-1} = S_f(w) \\ y_0 + 2y_1 + 3y_2 + \cdots + (m-1)y_{m-2} + 0 = S_{f+1}(w) \\ \vdots \\ (m-2)y_0 + (m-1)y_1 + 0 + \cdots + (m-4)y_{m-2} + (m-3)y_{m-1} = S_{f+m-2}(w) \\ (m-1)y_0 + 0 + y_2 + \cdots + (m-3)y_{m-2} + (m-2)y_{m-1} = S_{f+m-1}(w) \end{cases}$$

得到

$$y_i = \frac{1}{m}\{\boldsymbol{I}_{\{\mathbf{0}\}}(w) + [S_{f+m-1-i}(w) - S_{f+m-i}(w)]\}, \quad i = 0,1,\cdots,m-1$$

将之代入式(10.44)并注意到 $\sum_{i=0}^{m-1} u^{ki} = 0, k = 1,2,\cdots,m-1$ 即得式(10.41)。再将式(10.41)稍作变形即得

$$\begin{aligned} S_{(kf)}(w) &= \frac{1}{m}\left[\sum_{i=0}^{m-1} u^{ki} S_{f+m-1-i}(w) - \sum_{i=0}^{m-1} u^{ki} S_{f+m-i}(w)\right] \\ &= \frac{1}{m}\left[\sum_{\lambda=0}^{m-1} u^{k(m-\lambda-1)} S_{f+\lambda}(w) - \sum_{\lambda=0}^{m-1} u^{k(m-\lambda)} S_{f+\lambda}(w)\right] \\ &= \frac{1}{m}\sum_{\lambda=0}^{m-1}[u^{k(m-\lambda-1)} - u^{k(m-\lambda)}]S_{f+\lambda}(w) \\ &= \frac{1}{m}\sum_{\lambda=0}^{m-1}[u^{-k} - 1]u^{-k\lambda} S_{f+\lambda}(w) \end{aligned}$$

可见式(10.42)也成立。

根据式(10.40)和式(10.43),则有

$$\begin{aligned} \sum_{\lambda=0}^{m-1} S_{f+\lambda}(w) &= \sum_{\lambda=0}^{m-1}\sum_{i=0}^{m-1}(i+\lambda)_{(\mathrm{mod}\ m)} y_i = \sum_{i=0}^{m-1} y_i \sum_{\lambda=0}^{m-1}[(i+\lambda)_{(\mathrm{mod}\ m)}] \\ &= \sum_{i=0}^{m-1} y_i[1 + 2 + \cdots + (m-1)] \\ &= \begin{cases} \dfrac{m(m-1)}{2} & w = 0 \\ 0 & w \neq 0 \end{cases} \end{aligned}$$

下面的定理给出了用 m 值逻辑函数

$$kf(x), \quad x \in Z_m^n, k = 1,2,\cdots,m-1$$

的循环 Chrestenson 谱 $S_{(kf)}(w), w \in Z_m^n$ 线性表出 m 值逻辑函数

$$f(x) + \lambda, \quad x \in Z_m^n, \quad \lambda = 0,1,\cdots,m-1$$

的线性 Chrestenson 谱 $S_{f+\lambda}(w), S_{f+\lambda}(w), w \in Z_m^n$ 的关系式,这里采用的同样是解线

性方程组的简单方法。

定理 10.2.8　对任意的 n 元 m 值逻辑函数 $f(x)$，$x\in Z_m^n$ 以及任意的 $w\in Z_m^n$ 和 $\lambda=0,1,\cdots,m-1$，都有

$$S_{f+\lambda}(w)=\sum_{i=0}^{m-1}(i+\lambda)_{(\mathrm{mod}\ m)}S_i \tag{10.45}$$

其中

$$S_i=\frac{1}{m}\boldsymbol{I}_{\{0\}}(w)+\frac{1}{m}\sum_{k=1}^{m-1}u^{-ki}S_{(kf)}(w) \tag{10.46}$$

因而

$$S_{f+\lambda}(w)=\frac{m-1}{2}\boldsymbol{I}_{\{0\}}(w)+\sum_{k=1}^{m-1}\frac{u^{(\lambda+1)k}}{1-u^k}S_{(kf)}(w) \tag{10.47}$$

证明：由式(10.44)可得以下关于

$$y_i=\sum_{j=0}^{m-1}u^jP\{f(X)=i,w^*\cdot X=j\},\quad i=0,1,\cdots,m-1$$

的方程组

$$\begin{cases}y_0+uy_1+u^2y_2+\cdots+u^{m-1}y_{m-1}=S_{(f)}(w)\\ y_0+u^2y_1+u^4y_2+\cdots+u^{2(m-1)}y_{m-1}=S_{(2f)}(w)\\ \vdots\\ y_0+u^ky_1+u^{2k}y_2+\cdots+u^{(m-1)k}y_{m-1}=S_{(kf)}(w)\\ \vdots\\ y_0+u^{m-1}y_1+u^{2(m-1)}y_2+\cdots+u^{(m-1)(m-1)}y_{m-1}=S_{((m-1)f)}(w)\end{cases}$$

注意到 i、$k=0,1,\cdots,m-1$ 时，成立

$$1+u^k\cdot u^{(m-i)}+u^{2k}\cdot u^{2(m-i)}+\cdots+u^{(m-1)k}\cdot u^{(m-1)(m-i)}=\begin{cases}m & k=i\\ 0 & k\neq i\end{cases}$$

依次在以上第 $k(k=0,1,\cdots,m-1)$个方程的两端同乘以 $u^{k(m-i)}$ 再相加即得

$$(m-1)y_i-\sum_{j\neq i}y_j=\sum_{k=1}^{m-1}u^{k(m-i)}S_{(kf)}(w),\quad i=0,1,\cdots,m-1$$

由此，据定理 10.2.6 有

$$\begin{aligned}y_i&=\frac{1}{m}\sum_{j=0}^{m-1}y_j+\frac{1}{m}\sum_{k=1}^{m-1}u^{k(m-i)}S_{(kf)}(w)\\ &=\frac{1}{m}\boldsymbol{I}_{\{0\}}(w)+\frac{1}{m}\sum_{k=1}^{m-1}u^{k(m-i)}S_{(kf)}(w),\quad i=0,1,\cdots,m-1\end{aligned}$$

将之代入式(10.43)即可得式(10.45)

$$\begin{aligned}\sum_{l=0}^{m-1}l\cdot u^{-lk}&=a\times\frac{\mathrm{d}}{\mathrm{d}x}(1+x+x^2+\cdots+x^{m-1})\Big|_{x=a}\\ &=a\times\frac{\mathrm{d}}{\mathrm{d}x}\left(\frac{1-x^m}{1-x}\right)\Big|_{x=a}\\ &=a\times\left[\frac{-mx^{m-1}(1-x)-(1-x^m)(-1)}{(1-x)^2}\right]\Big|_{x=a}\end{aligned}$$

$$=a\times\left[\frac{-ma^{m-1}+ma^{m}+1-a^{m}}{(1-a)^2}\right]$$

$$=a\times\frac{m(1-a^{-1})}{(1-a)^2}$$

$$=\frac{m}{a-1}=\frac{m}{u^{-k}-1}=\frac{mu^k}{1-u^k},\quad k=1,2,\cdots,m-1$$

又可有

$$S_{f+\lambda}(w)=\sum_{i=0}^{m-1}(i+\lambda)_{(\bmod m)}S_i$$

$$=\frac{1}{m}\boldsymbol{I}_{\{0\}}(w)\sum_{i=0}^{m-1}[(i+\lambda)_{(\bmod m)}]$$

$$+\frac{1}{m}\sum_{k=1}^{m-1}S_{(kf)}(w)\cdot u^{k\lambda}\sum_{i=0}^{m-1}[(i+\lambda)_{(\bmod m)}]\cdot u^{-k(i+\lambda)}$$

$$=\frac{m-1}{2}\boldsymbol{I}_{\{0\}}(w)+\frac{1}{m}\sum_{k=1}^{m-1}S_{(kf)}(w)\cdot u^{k\lambda}\sum_{l=0}^{m-1}l\cdot u^{-lk}$$

$$=\frac{m-1}{2}\boldsymbol{I}_{\{0\}}(w)+\frac{1}{m}\sum_{k=1}^{m-1}S_{(kf)}(w)\cdot u^{k\lambda}\cdot\frac{mu^k}{1-u^k}$$

$$=\frac{m-1}{2}\boldsymbol{I}_{\{0\}}(w)+\sum_{k=1}^{m-1}\frac{u^{(\lambda+1)k}}{1-u^k}S_{(kf)}(w),\quad \lambda=0,1,\cdots,m-1$$

即式(10.47)也成立。

定理 10.2.7 和定理 10.2.8 表明，m 值逻辑函数的两种 Chrestenson 谱能够相互线性表出，因而 m 值逻辑函数的凡能用一种 Chrestenson 谱予以刻画的性质用另一种 Chrestenson 谱也能刻画，这为研究问题带来极大方便。

例 10.2.5 设 2 元 3 值逻辑函数

$$f(x_1,x_2)=x_1x_2,\quad (x_1,x_2)\in Z_3^2$$

验证两种 Chrestenson 谱之间的关系式成立。

解：由 Chrestenson 谱的定义有：

(1) $f(x_1,x_2)$的线性和循环 Chrestenson 谱分别为

$$S_f(0,0)=\frac{2}{3},\quad S_f(0,1)=S_f(0,2)=S_f(1,0)=S_f(2,0)=-\frac{1}{3}$$

$$S_f(1,1)=S_f(2,2)=\frac{1}{3},\quad S_f(1,2)=S_f(2,1)=0$$

和

$$S_{(f)}(0,0)=S_{(f)}(0,1)=S_{(f)}(1,0)=S_{(f)}(0,2)=S_{(f)}(2,0)=\frac{1}{3}$$

$$S_{(f)}(1,1)=\frac{1}{3}u^2,\quad S_{(f)}(1,2)=S_{(f)}(2,1)=\frac{1}{3}u,\quad S_{(f)}(2,2)=\frac{1}{3}u^2$$

(2) $f(x_1,x_2)+1$ 的线性 Chrestenson 谱为

$$S_{f+1}(0,0)=1,\quad S_{f+1}(0,1)=S_{f+1}(0,2)=S_{f+1}(1,0)=S_{f+1}(2,0)=0$$

$$S_{f+1}(1,1)=S_{f+1}(2,2)=-\frac{1}{3},\quad S_{f+1}(1,2)=S_{f+1}(2,1)=\frac{1}{3}$$

(3) $f(x_1,x_2)+2$ 的线性 Chrestenson 谱为

$$S_{f+2}(0,0)=\frac{4}{3},\quad S_{f+2}(0,1)=S_{f+2}(0,2)=S_{f+2}(1,0)=S_{f+2}(2,0)=\frac{1}{3}$$

$$S_{f+2}(1,2)=S_{f+2}(2,1)=-\frac{1}{3},\quad S_{f+2}(1,1)=S_{f+2}(2,2)=0$$

(4) $2f(x_1,x_2)$的循环 Chrestenson 谱为

$$S_{(2f)}(0,0)=S_{(2f)}(0,1)=S_{(2f)}(1,0)=S_{(2f)}(0,2)=S_{(2f)}(2,0)=\frac{1}{3}$$

$$S_{(2f)}(1,1)=\frac{1}{3}u,\quad S_{(2f)}(1,2)=S_{(2f)}(2,1)=\frac{1}{3}u^2,\quad S_{(2f)}(2,2)=\frac{1}{3}u$$

由式(10.42)

$$S_{(f)}(w)=\frac{u^2-1}{3}(S_f(w)+u^2S_{f+1}(w)+uS_{f+2}(w))$$

$$S_{(2f)}(w)=\frac{u-1}{3}(S_f(w)+uS_{f+1}(w)+u^2S_{f+2}(w))$$

将上述值代入即可得结论。如 $w=(1,1)$时

$$S_f(1,1)=\frac{1}{3},\quad S_{f+1}(1,1)=-\frac{1}{3},\quad S_{f+2}(1,1)=0,$$

$$S_{(f)}(1,1)=\frac{1}{3}u^2,\quad S_{(2f)}(1,1)=\frac{1}{3}u$$

$$S_{(f)}(1,1)=\frac{u^2-1}{3}(S_f(1,1)+u^2S_{f+1}(1,1)+uS_{f+2}(1,1))$$

$$=\frac{u^2-1}{3}\left(\frac{1}{3}-\frac{1}{3}u^2\right)=\frac{u^2}{3}$$

$$S_{(2f)}(1,1)\frac{u-1}{3}=(S_f(1,1)+uS_{f+1}(1,1)+u^2S_{f+2}(1,1))$$

$$=\frac{u-1}{3}\left(\frac{1}{3}-\frac{1}{3}u\right)=\frac{u}{3}$$

由式(10.47)

$$S_f(w)=\boldsymbol{I}_{\{0\}}(w)+\frac{u}{1-u}S_{(f)}(w)+\frac{u^2}{1-u^2}S_{(2f)}(w)$$

$$S_{f+1}(w)=\boldsymbol{I}_{\{0\}}(w)+\frac{u^2}{1-u}S_{(f)}(w)+\frac{u}{1-u^2}S_{(2f)}(w)$$

$$S_{f+2}(w)=\boldsymbol{I}_{\{0\}}(w)+\frac{1}{1-u}S_{(f)}(w)+\frac{1}{1-u^2}S_{(2f)}(w)$$

当将上述值代入即可得结论。如 $w=(1,1)$时

$$S_f(1,1)=\frac{1}{3},\quad S_{f+1}(1,1)=-\frac{1}{3},\quad S_{f+2}(1,1)=0,$$

$$S_{(f)}(1,1)=\frac{1}{3}u^2,\quad S_{(2f)}(1,1)=\frac{1}{3}u$$

$$S_f(1,1)=\frac{u}{1-u}S_{(f)}(1,1)+\frac{u^2}{1-u^2}S_{(2f)}(1,1)$$
$$=\frac{u}{1-u}\times\frac{1}{3}u^2+\frac{u^2}{1-u^2}\times\frac{1}{3}u=\frac{1}{3}$$
$$S_{f+1}(1,1)=\frac{u^2}{1-u}S_{(f)}(1,1)+\frac{u}{1-u^2}S_{(2f)}(1,1)$$
$$=\frac{u^2}{1-u}\times\frac{1}{3}u^2+\frac{u}{1-u^2}\times\frac{1}{3}u=-\frac{1}{3}$$
$$S_{f+2}(1,1)=\frac{1}{1-u}S_{(f)}(1,1)+\frac{1}{1-u^2}S_{(2f)}(1,1)$$
$$=\frac{1}{1-u}\times\frac{1}{3}u^2+\frac{1}{1-u^2}\times\frac{1}{3}u=0$$

10.2.4　Chrestenson 谱的快速计算

在 Chrestenson 谱的研究与应用中，其快速计算是一个非常重要的问题，因为它关系到其应用的有效性问题。本节将简要介绍一种计算 Chrestenson 谱的快速算法。

为了讨论方便起见，假定 $m=p$ 为素数，$\boldsymbol{A}_n=(u^{w\cdot x})_{p^n\times p^n}$，$w,x\in Z_p^n$，$w$ 和 x 以字典序取值。

例 10.2.6　当 $n=1,p=3$ 时，则

$$\boldsymbol{A}_1=\begin{pmatrix}1&1&1\\1&u&u^2\\1&u^2&u\end{pmatrix}$$

为了叙述方便，将向量$(x_1,x_2,\cdots,x_n)\in Z_p^n$ 用其对应的整数 $x(0\leqslant x\leqslant p^n-1)$来表示。设

$$\boldsymbol{f}=(f(0),f(1),\cdots,f(2^n-1))$$
$$\boldsymbol{u}^f=(u^{f(0)},u^{f(1)},\cdots,u^{f(p^n-1)})$$
$$\boldsymbol{S}_f=(S_f(0),S_f(1),\cdots,S_f(2^n-1))$$
$$\boldsymbol{S}_{(f)}=(S_{(f)}(0),S_{(f)}(1),\cdots,S_{(f)}(2^n-1))$$

由 Chrestenson 谱的定义式(10.29)和式(10.30)可知

$$\boldsymbol{S}_f=\frac{1}{p^n}\boldsymbol{f}\,\overline{\boldsymbol{A}}_n \tag{10.48}$$

$$\boldsymbol{S}_{(f)}=\frac{1}{p^n}\boldsymbol{u}^f\,\overline{\boldsymbol{A}}_n \tag{10.49}$$

其中$\overline{\boldsymbol{A}}_n$表示 $\boldsymbol{A}_n$ 的复共轭即$\overline{\boldsymbol{A}}_n=(u^{-w\cdot x})_{p^n\times p^n}$。

对给定的 $w=(w_1,w_2,\cdots,w_n)$，置 $\overleftarrow{w}=(w_n,\cdots,w_2,w_1)$。对给定的 $p^n\times p^n$ 阶对称矩阵 $\boldsymbol{A}_n$，置$\overleftarrow{\boldsymbol{A}}_n$表示满足以下条件的 $p^n\times p^n$ 阶对称矩阵：$\overleftarrow{\boldsymbol{A}}_n$的第 w 行(列)是 $\boldsymbol{A}_n$ 的第 $\overleftarrow{w}$ 行(列)，令

$$
\boldsymbol{C}_n = \begin{pmatrix}
e_0^0 e_1^0 \cdots e_{p-1}^0 & & & \\
 & e_0^0 e_1^0 \cdots e_{p-1}^0 & & \\
 & & \ddots & \\
 & & & e_0^0 e_1^0 \cdots e_{p-1}^0 \\
e_0^1 e_1^1 \cdots e_{p-1}^1 & & & \\
 & e_0^1 e_1^1 \cdots e_{p-1}^1 & & \\
 & & \ddots & \\
 & & & e_0^1 e_1^1 \cdots e_{p-1}^1 \\
\cdots & \cdots & \cdots & \cdots \\
e_0^{p-1} e_1^{p-1} \cdots e_{p-1}^{p-1} & & & \\
 & e_0^{p-1} e_1^{p-1} \cdots e_{p-1}^{p-1} & & \\
 & & \ddots & \\
 & & & e_0^{p-1} e_1^{p-1} \cdots e_{p-1}^{p-1}
\end{pmatrix}
$$

则可以证明

$$
\boldsymbol{A}_n = \overleftarrow{(\boldsymbol{C}_n)^n} \tag{10.50}
$$

其中 $e_t = u^t, t = 0, 1, \cdots, p-1$。

利用 $\boldsymbol{A}_n$ 的分解式(10.50)及式(10.48)和式(10.49)可有效地计算 Chrestenson 谱。

例 10.2.7　设 2 元 3 值逻辑函数

$$
f(x_1, x_2) = x_1 x_2, \quad (x_1, x_2) \in Z_3^2
$$

利用快速算法计算 Chrestenson 谱。

解：

$$
\boldsymbol{A}_2 = \begin{pmatrix}
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & u & u^2 & 1 & u & u^2 & 1 & u & u^2 \\
1 & u^2 & u & 1 & u^2 & u & 1 & u^2 & u \\
1 & 1 & 1 & u & u & u & u^2 & u^2 & u^2 \\
1 & u & u^2 & u & u^2 & 1 & u^2 & 1 & u \\
1 & u^2 & u & u & 1 & u^2 & u^2 & u & 1 \\
1 & 1 & 1 & u^2 & u^2 & u^2 & u & u & u \\
1 & u & u^2 & u^2 & 1 & u & u & u^2 & 1 \\
1 & u^2 & u & u^2 & u & 1 & u & 1 & u^2
\end{pmatrix}
$$

则

$$
\bar{\boldsymbol{A}}_2 = \begin{pmatrix}
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & u^2 & u & 1 & u^2 & u & 1 & u^2 & u \\
1 & u & u^2 & 1 & u & u^2 & 1 & u & u^2 \\
1 & 1 & 1 & u^2 & u^2 & u^2 & u & u & u \\
1 & u^2 & u & u^2 & u & 1 & u & 1 & u^2 \\
1 & u & u^2 & u^2 & 1 & u & u & u^2 & 1 \\
1 & 1 & 1 & u & u & u & u^2 & u^2 & u^2 \\
1 & u^2 & u & u & 1 & u^2 & u^2 & u & 1 \\
1 & u & u^2 & u & u^2 & 1 & u^2 & 1 & u
\end{pmatrix}
$$

$$\boldsymbol{f} = (f(0), f(1), \cdots, f(2^n - 1)) = (0,0,0,0,1,2,0,2,1)$$
$$\boldsymbol{u}^f = (u^{f(0)}, u^{f(1)}, \cdots, u^{f(p^n-1)}) = (1,1,1,1,u,u^2,1,u^2,u)$$
$$S_f = (S_f(0), S_f(1), \cdots, S_f(2^n - 1))$$
$$= \frac{1}{9}\boldsymbol{f}\bar{\boldsymbol{A}}_2 = \left(\frac{2}{3}, -\frac{1}{3}, -\frac{1}{3}, -\frac{1}{3}, \frac{1}{3}, 0, -\frac{1}{3}, 0, \frac{1}{3}\right)$$
$$\boldsymbol{S}_{(f)} = (S_{(f)}(0), S_{(f)}(1), \cdots, S_{(f)}(2^n - 1))$$
$$= \frac{1}{9}\boldsymbol{u}^f\bar{\boldsymbol{A}}_2 = \left(\frac{1}{3}, \frac{1}{3}, \frac{1}{3}, \frac{1}{3}, \frac{1}{3}u^2, \frac{1}{3}u, \frac{1}{3}, \frac{1}{3}u, \frac{1}{3}u^2\right)$$

10.2.5 m 值逻辑函数自相关函数的定义及其性质

如同布尔函数时的情况一样,下述结论在研究多值逻辑函数的有关性质和构造时也能发挥很好的作用。

定义 10.2.3 设 $f(x), x\in Z_m^n$ 是 n 元 m 值逻辑函数,对

$$x = (x_1, x_2, \cdots, x_n) \in Z_m^n, \quad s = (s_1, s_2, \cdots, s_n) \in Z_m^n$$

记 $x+s=(x_1+s_1, x_2+s_2, \cdots, x_n+s_n)$,称

$$r_f(s) = \frac{1}{m^n}\sum_{x\in Z_m^n} u^{f(x+s)-f(x)}, \quad s \in Z_m^n \tag{10.51}$$

为 n 元 m 值逻辑函数 $f(x), x\in Z_m^n$ 的自相关函数。

定义 10.2.4 设 $f_1(x), f_2(x), x\in Z_m^n$ 是两个 n 元 m 值逻辑函数,称

$$r_{f_1f_2}(s) = \frac{1}{m^n}\sum_{x\in Z_m^n} u^{f_1(x+s)-f_2(x)}, \quad s \in Z_m^n \tag{10.52}$$

为 n 元 m 值逻辑函数 $f_1(x)$ 和 $f_2(x)$ 的互相关函数。

例 10.2.8 设 4 值逻辑函数 $f(x_1, x_2) = x_1x_2 + 1, (x_1, x_2) \in Z_4^2$,试确定 $f(x_1, x_2)$ 的自相关函数。

解:由自相关函数的定义,则对任意的 $s=(s_1, s_2)\in Z_4^2$,有

$$f(x_1 + s_1, x_2 + s_2) + f(x_1, x_2) = (x_1 + s_1)(x_2 + s_2) + 1 + x_1x_2 + 1$$
$$= s_2x_1 + s_1x_2 + s_1s_2$$

$$r_f(0,0) = 1,$$
$$r_f(0,1) = \frac{1}{4^2}\sum_{x\in Z_4^2} i^{f(x+s)-f(x)} = \frac{1}{4^2}(4i^0 + 4i^1 + 4i^2 + 4i^3) = 0$$

由对称性,有

$$r_f(0,1) = r_f(1,0) = r_f(0,3) = r_f(3,0) = 0$$
$$r_f(0,2) = \frac{1}{4^2}\sum_{x\in Z_4^2} i^{f(x+s)-f(x)} = \frac{1}{4^2}(8i^0 + 8i^2) = 0$$

由对称性,有

$$r_f(0,2) = r_f(2,0) = 0$$

当 $\gcd(s_1, s_2)=1$ 时,$s_2x_1 + s_1x_2 + s_1s_2$ 是平衡的,则 $r_f(s_1, s_2)=0$,即

$$r_f(1,1) = r_f(1,2) = r_f(1,3) = r_f(2,1) = r_f(2,3)$$

$$=r_f(3,1)=r_f(3,2)=0$$

当 $\gcd(s_1,s_2)=2$ 时，$s_2x_1+s_1x_2+s_1s_2$ 只取 0 和 2，且当 x 跑遍 Z_4^2 时，只取 0 和 2 的个数相等，则 $r_f(s_1,s_2)=0$，即

$$r_f(2,2)=0$$

当 $(s_1,s_2)=(3,3)$ 时，$s_2x_1+s_1x_2+s_1s_2$ 是平衡的，则 $r_f(3,3)=0$。

下面定理刻画了 m 值逻辑函数的自相关函数与其 Chrestenson 循环谱之间的关系。

定理 10.2.9　设 $f(x),x\in Z_m^n$ 是 m 值逻辑函数，其自相关函数和 Chrestenson 循环谱分别为 $r_f(s)$ 和 $S_{(f)}(w)$，则

$$\frac{1}{m^n}\sum_{s\in Z_m^n}r_f(s)\cdot u^{-w\cdot s}=|S_{(f)}(w)|^2,\quad w\in Z_m^n \tag{10.53}$$

$$\sum_{w\in Z_m^n}|S_{(f)}(w)|^2\cdot u^{w\cdot s}=r_f(s),\quad s\in Z_m^n \tag{10.54}$$

又设 $f_1(x),f_2(x),x\in Z_m^n$ 是两个 m 值逻辑函数，其互相关函数为 $r_{f_1f_2}(s)$，则

$$\frac{1}{m^n}\sum_{s\in Z_m^n}r_{f_1f_2}(s)\cdot u^{-w\cdot s}=S_{(f_1)}(w)\cdot\overline{S_{(f_2)}(w)},\quad w\in Z_m^n \tag{10.55}$$

$$\sum_{w\in Z_m^n}S_{(f_1)}(w)\cdot\overline{S_{(f_2)}(w)}\cdot u^{w\cdot s}=r_{f_1f_2}(s),\quad s\in Z_m^n \tag{10.56}$$

证明：根据 m 值逻辑函数自相关函数的定义知，对任意的 $w\in Z_m^n$，有

$$\begin{aligned}\frac{1}{m^n}\sum_{s\in Z_m^n}r_f(s)\cdot u^{-w\cdot s}&=\frac{1}{m^n}\sum_{s\in Z_m^n}\left(\frac{1}{m^n}\sum_{x\in Z_m^n}u^{f(x+s)-f(x)}\right)\cdot u^{-w\cdot s}\\&=\frac{1}{m^{2n}}\sum_{x\in Z_m^n}u^{-f(x)}\sum_{s\in Z_m^n}u^{f(x+s)-w\cdot s}\\&=\frac{1}{m^{2n}}\sum_{x\in Z_m^n}u^{-f(x)}\sum_{y\in Z_m^n}u^{f(y)-w\cdot(y-x)}\\&=\left(\frac{1}{m^n}\sum_{x\in Z_m^n}u^{-f(x)+w\cdot x}\right)\cdot\left(\frac{1}{m^n}\sum_{y\in Z_m^n}u^{f(y)-w\cdot y}\right)\\&=\overline{S_{(f)}(w)}\cdot S_{(f)}(w)=|S_{(f)}(w)|^2\end{aligned}$$

因而式(10.53)成立。

$$\begin{aligned}\sum_{w\in Z_m^n}|S_{(f)}(w)|^2\cdot u^{w\cdot s}&=\sum_{w\in Z_m^n}\left[\frac{1}{m^n}\sum_{v\in Z_m^n}r_f(v)\cdot u^{-w\cdot v}\right]\cdot u^{w\cdot s}\\&=\frac{1}{m^n}\sum_{v\in Z_m^n}r_f(v)\sum_{w\in Z_m^n}u^{w\cdot(s-v)}\\&=\frac{1}{m^n}r_f(s)\sum_{w\in Z_m^n}u^{w\cdot(s-s)}+\frac{1}{m^n}\sum_{v\in Z_m^n,v\neq s}r_f(v)\sum_{w\in Z_m^n}u^{w\cdot(s-v)}\\&=r_f(s),\quad s\in Z_m^n\end{aligned}$$

同样可证式(10.55)和式(10.56)。

10.2.6 Chrestenson 谱的应用举例

1. m 值逻辑函数的最佳仿射逼近

定理 10.2.10 对任一 n 元 m 值逻辑函数 $f(x), x\in Z_m^n$ 及任一 $w\in Z_m^n$ 和 $a\in Z_m$，都有

$$P\{f(X)=w\cdot X+a\}=\frac{1}{m}+\frac{1}{m}\sum_{k=1}^{m-1}u^{-ka}S_{(kf)}(kw) \qquad (10.57)$$

证明：注意到对任一 $w\in Z_m^n$ 和 $k=1,2,\cdots,m-1$，都成立

$$S_{(kf)}(kw)=\frac{1}{m^n}\sum_{x\in Z_m^n}u^{kf(x)-kw\cdot x}=\frac{1}{m^n}\sum_{x\in Z_m^n}u^{k[f(x)-w\cdot x]}$$
$$=\sum_{a=0}^{m-1}u^{ka}P\{f(X)=w\cdot X+a\} \qquad (10.58)$$

记

$$y_a=P\{f(X)=w\cdot X+a\},\quad a\in Z_m$$

则由概率的性质和式(10.58)可得

$$\begin{cases}y_0+y_1+y_2+\cdots+y_{m-1}=1\\ y_0+uy_1+u^2y_2+\cdots+u^{m-1}y_{m-1}=S_{(f)}(w)\\ \vdots\\ y_0+u^ky_1+u^{2k}y_2+\cdots+u^{(m-1)k}y_{m-1}=S_{(kf)}(kw)\\ \vdots\\ y_0+u^{m-1}y_1+u^{2(m-1)}y_2+\cdots+u^{(m-1)(m-1)}y_{m-1}=S_{((m-1)f)}(w)\end{cases}$$

解上述关于 $y_a=P\{f(X)=w\cdot X+a\}, a\in Z_m$ 的方程组即知式(10.57)成立。

由定理 10.2.10 立即得到下面多值逻辑函数的最佳仿射逼近谱表示定理。

定理 10.2.11 对任一 m 值逻辑函数 $f(x), x\in Z_m^n$，若 $w^*\in Z_m^n$ 和 $a^*\in Z_m$，使得

$$\sum_{k=1}^{m-1}u^{-ka^*}S_{(kf)}(kw^*)=\max\left\{\sum_{k=1}^{m-1}u^{-ka}S_{(kf)}(kw): w\in Z_m^n, a\in Z_m\right\}$$

则 $w^*\cdot x+a^*, x\in Z_m^n$ 即为 m 值逻辑函数 $f(x), x\in Z_m^n$ 的最佳仿射逼近。

2. *Chrestenson* 谱在 m 值逻辑函数相关免疫性判别中的应用

定义 10.2.5 设 $f(x_1,x_2,\cdots,x_n),(x_1,x_2,\cdots,x_n)\in Z_m^n$ 是 n 元 m 值逻辑函数，若对任意的 $1\leqslant i_1<\cdots<i_k\leqslant n$，$m$ 值随机变量 $f(X_1,X_2,\cdots,X_n)$ 与 m 值随机向量 $(X_{i_1},X_{i_2},\cdots,X_{i_k})$ 都相互独立，则称 m 值逻辑函数 $f(x_1,x_2,\cdots,x_n),(x_1,x_2,\cdots,x_n)\in Z_m^n$ 是 k 阶相关免疫的。

同布尔函数一样，给出以下谱判别定理。

定理 10.2.12 m 值逻辑函数 $f(x_1,x_2,\cdots,x_n),(x_1,x_2,\cdots,x_n)\in Z_m^n$ 为 k 阶相关免疫的充要条件是对任意 $w\in Z_m^n$，$1\leqslant W_H(w)\leqslant k$ 和任意正整数 $\lambda\mid N, \lambda\neq N$，都有

$$S_{(\lambda f)}(w)=0 \qquad (10.59)$$

定理 10.2.12 的证明可参见文献[4]。

例 10.2.9　尽管 2 元 6 值逻辑函数

$$f(x_1,x_2)=2x_1+3x_2,\quad x_1\in Z_6,x_2\in Z_6$$

的 Chrestenson 循环谱满足：对 $(w_1,w_2)\in Z_6^2$，当 $W(w_1,w_2)=1$ 时，就有 $S_{(f)}(w_1,w_2)=0$。但它并不具有 1 阶相关免疫性。

证明：对 $(w_1,w_2)\in Z_6^2$，当 $W(w_1,w_2)=1$ 时，若 $w_1\neq 0$ 而 $w_2=0$，则有

$$\begin{aligned}S_{(f)}(w_1,w_2)&=\frac{1}{6^2}\sum_{x_1\in Z_6}\sum_{x_2\in Z_6}u^{2x_1+3x_2-w_1x_1}=\frac{1}{6^2}\sum_{x_1\in Z_6}u^{(2-w_1)x_1}\sum_{x_2\in Z_6}u^{3x_2}\\&=\frac{1}{6^2}[1+u^3+u^6+\cdots+u^{15}]\sum_{x_1\in Z_6}u^{(2-w_1)x_1}\\&=\frac{1}{6^2}\times\frac{1-(u^3)^6}{1-u^3}\sum_{x_1\in Z_6}u^{(2-w_1)x_1}\\&=\frac{1}{6^2}\times\frac{1-(u^6)^3}{1-u^3}\sum_{x_1\in Z_6}u^{(2-w_1)x_1}=0\end{aligned}$$

若 $w_1=0$ 而 $w_2\neq 0$，则有

$$\begin{aligned}S_{(f)}(w_1,w_2)&=\frac{1}{6^2}\sum_{x_1\in Z_6}\sum_{x_2\in Z_6}u^{2x_1+3x_2-w_3x_2}=\frac{1}{6^2}\sum_{x_2\in Z_6}u^{(3-w_3)x_1}\sum_{x_1\in Z_6}u^{2x_1}\\&=\frac{1}{6^2}[1+u^2+u^4+\cdots+u^{10}]\sum_{x_1\in Z_6}u^{(3-w_2)x_2}\\&=\frac{1}{6^2}\times\frac{1-(u^2)^6}{1-u^2}\sum_{x_2\in Z_6}u^{(3-w_2)x_2}\\&=\frac{1}{6^2}\times\frac{1-(u^6)^2}{1-u^2}\sum_{x_2\in Z_6}u^{(3-w_2)x_2}=0\end{aligned}$$

但是，由于

$$P\{2X_1+3X_2=j\}=\frac{1}{6},\quad j\in Z_4$$

$$P\{2X_1+3X_2=1,X_2=0\}$$

$$=P\{2X_2=1,X_2=0\}=0\neq\frac{1}{36}=P\{2X_1+3X_2=1\}\cdot P\{X_2=0\}$$

可见 $2X_1+3X_2$ 和 X_2 不相互独立，故 $f(x_1,x_2)=2x_1+3x_2,x_1\in Z_6,x_2\in Z_6$ 是不具有 1 阶相关免疫性的。事实上，此时

$$S_{(2f)}(4,0)=\sum_{j=0}^{3}u^jP\{2(2X_1+3X_2)=4X_1+j\}=1$$

即函数 $f(x_1,x_2)=2x_1+3x_2,x_1\in Z_6,x_2\in Z_6$ 确实不满足定理 10.2.12 中的充分条件。

3. *Chrestenson* 谱在扩散特性研究中的应用

定义 10.2.6　若 n 元 m 值逻辑函数 $f(x),x\in Z_m^n$ 的自相关函数 $r_f(\cdot)$ 满足

$$r_f(s)=0,\quad s\in Z_m^n,\quad W_H(s)=1\tag{10.60}$$

则称 $f(x),x\in Z_m^n$ 是满足严格雪崩准则的。

定义 10.2.7 设 $f(x),x\in Z_m^n$ 是一个 n 元 m 值逻辑函数。

(1) 对于 $s\in Z_m^n$，若 $f(x)$ 的自相关函数 $r_f(\cdot)$ 满足 $r_f(s)=0$，则称 $f(x),x\in Z_m^n$ 关于 $s\in Z_m^n$ 是满足扩散准则的。

(2) 若对所有的 $s\in Z_m^n$，当 $1\leqslant W_H(s)\leqslant k$ 时，$f(x)$ 的自相关函数 $r_f(\cdot)$ 满足

$$r_f(s)=0 \tag{10.61}$$

则称 $f(x),x\in Z_m^n$ 是满足 k 次扩散准则的。

例 10.2.10 设 4 值逻辑函数 $f(x_1,x_2)=x_1x_2+1,(x_1,x_2)\in Z_4^2$，由例 10.2.8 的求解过程可知 $f(x_1,x_2)$ 满足 2 次扩散准则。

根据 m 值逻辑函数的自相关函数和 Chrestenson 谱的关系容易得到满足严格雪崩准则和扩散准则的 m 值逻辑函数的 Chrestenson 谱的以下性质。

定理 10.2.13 m 值逻辑函数 $f(x),x\in Z_m^n$ 满足严格雪崩准则的充分必要条件是对所有的 $1\leqslant j\leqslant n$，只要 $s_j\in Z_m\backslash\{0\}$，就有

$$\sum_{w\in Z_m^n}|S_{(f)}(w)|^2\cdot u^{w_j\cdot s_j}=0 \tag{10.62}$$

再由 m 值逻辑函数满足严格雪崩准则的定义即得以下定理。

定理 10.2.14 m 值逻辑函数 $f(x),x\in Z_m^n$ 关于 $s\in Z_m^n$ 满足扩散准则的充分必要条件是

$$\sum_{w\in Z_m^n}|S_{(f)}(w)|^2\cdot u^{w\cdot s}=0 \tag{10.63}$$

而 m 值逻辑函数 $f(x),x\in Z_m^n$ 满足 k 次扩散准则的充分必要条件是对的所有 $s\in Z_m^n$，当 $1\leqslant W_H(s)\leqslant k$ 时，都有

$$\sum_{w\in Z_m^n}|S_{(f)}(w)|^2\cdot u^{w\cdot s}=0 \tag{10.64}$$

10.3 有限域上的频谱方法与技术

10.3.1 有限域上的离散傅里叶变换技术

设 α 是有限域 F_q 的某一扩域 F_q^m 中的一个 N 阶元素。一个 F_q 上的序列 $a^N=a_0a_1\cdots a_{N-1}$（也称为“时域”(time-domain)序列）的离散傅里叶变换（简记为 DFT），定义为一个 F_q^m 上的序列 $A^N=A_0A_1\cdots A_{N-1}$（也称为“频域”(frequency-domain)序列），其中

$$A_i=\sum_{j=0}^{N-1}a_j\alpha^{ij},\quad i=0,1,\cdots,N-1 \tag{10.65}$$

逆 DFT 为

$$a_j=\frac{1}{N^*}\sum_{i=0}^{N-1}A_i\alpha^{-ij},\quad j=0,1,\cdots,N-1 \tag{10.66}$$

其中 $N^*=N \bmod p$，p 是 F_q 的特征。

令

$$F = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & \alpha & \alpha^2 & \cdots & \alpha^{N-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \alpha^{N-1} & \alpha^{2(N-1)} & \cdots & \alpha^{(N-1)(N-1)} \end{bmatrix}$$

则 $\boldsymbol{A}^N$ 可表示为

$$\boldsymbol{A}^N = \alpha^N \boldsymbol{F} \tag{10.67}$$

则 α^N 可表示为

$$\alpha^N = \frac{1}{N^*}\boldsymbol{A}^N \boldsymbol{F}^{-1} \tag{10.68}$$

这里

$$\boldsymbol{F}^{-1} = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & \alpha^{-1} & \alpha^{-2} & \cdots & \alpha^{-(N-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \alpha^{-(N-1)} & \alpha^{-2(N-1)} & \cdots & \alpha^{-(N-1)(N-1)} \end{bmatrix}$$

如果对所有的 i，按式(10.65)来定义 A_i，则有

$$A_{i+N} = \sum_{j=0}^{N-1} a_j \alpha^{(i+N)j} = \sum_{j=0}^{N-1} a_j \alpha^{ij} = A_i$$

于是，V^∞ 是周期为 N 的序列。如果对所有的 i，按式(10.66)来定义 a_i，则同样可以证明 a^∞ 是周期为 N 的序列。因此，可将 a^N 和 A^N 看成是由式(10.66)和式(10.65)分别定义的周期序列 a^∞ 和 A^∞ 的第一个周期段。

下面就来介绍一个很重要的定理——Blahut 定理。在叙述该定理之前，先介绍两个引理，这两个引理刻画了序列的线性复杂度和矩阵的秩之间的内在联系。

引理 10.3.1 设 $a^n = a_0 a_1 \cdots a_{n-1}$ 是 F_q 上的一个 n 长序列，则 a^n 的线性复杂度等于下面矩阵的最小秩：

$$\boldsymbol{G}_{a^n} = \begin{bmatrix} a_0 & a_1 & a_2 & \cdots & a_{n-3} & a_{n-2} & a_{n-1} \\ a_1 & a_2 & a_3 & \cdots & a_{n-2} & a_{n-1} & * \\ a_2 & a_3 & a_4 & \cdots & a_{n-1} & * & * \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ a_{n-1} & * & * & \cdots & * & * & * \end{bmatrix} \tag{10.69}$$

这里 * 为 F_q 中任意元素，且最小值是对这些任意元素赋所有可能值而求的。

证明：约定 $\boldsymbol{G}_{a^n}$ 最上面的行为第 0 行。如果 $L(a^n)=L$，则由线性复杂度的定义知，对于 $L\leqslant i\leqslant n-1$，只要适当地选择这些任意元素，$\boldsymbol{G}_{a^n}$ 的第 i 行可表示成前面 L 行的线性组合。因此，$\boldsymbol{G}_{a^n}$ 的最小秩不超过 L。但又由线性复杂度的定义知，L 是产生 a^n 的最短 LFSR 的级数，因而 L 是第 L 行的前 $n-L$ 个分量能够表示成前面行的对应分量线性组合的最小整数。这说明前 L 行线性独立，从而 $\boldsymbol{G}_{a^n}$ 的最小秩不小于 L，故 $\boldsymbol{G}_{a^n}$ 的最小秩等于 L。

引理 10.3.2 设 $a^\infty = a_0 a_1 \cdots a_{N-1} \cdots$ 是 F_q 上的一个周期为 N 的序列，则 a^∞ 的

线性复杂度等于下面循环矩阵的秩：

$$M(a^{\infty}) = \begin{bmatrix} a_0 & a_1 & \cdots & a_{N-1} \\ a_1 & a_2 & \cdots & a_0 \\ \vdots & \vdots & \ddots & \vdots \\ a_{N-1} & a_0 & \cdots & a_{N-2} \end{bmatrix} \tag{10.70}$$

引理 10.3.2 的证明类似于引理 10.3.1 的证明，故略。

定理 10.3.1（Blahut 定理） 周期为 N 的半无限“时域”序列 a^{∞} 的线性复杂度等于有限长“频域”序列 A^N 的汉明重量，其中，A^N 是 a^N 的 DFT，$L(a^{\infty})=W_H(A^N)$。类似地，周期为 N 的半无限序列 A^{∞} 的线性复杂度等于有限长序列 a^N 的汉明重量，其中，a^N 是 A^N 的 DFT，$L(A^{\infty})=W_H(a^N)$。

证明：由引理 10.3.1 知，a^{∞} 的线性复杂度等于下面循环矩阵的秩，即

$$M(a^{\infty}) = \begin{bmatrix} a_0 & a_1 & \cdots & a_{N-1} \\ a_1 & a_2 & \cdots & a_0 \\ \vdots & \vdots & \ddots & \vdots \\ a_{N-1} & a_0 & \cdots & a_{N-2} \end{bmatrix}$$

$M(a^{\infty})$ 可写成

$$M(a^{\infty}) = \frac{1}{N^*}FD_AF^{-1}$$

其中

$$D_A = \begin{bmatrix} A_0 & & & \\ & A_1 & & \\ & & \ddots & \\ & & & A_{N-1} \end{bmatrix}$$

因为 F 和 F^{-1} 互为逆矩阵，故秩 $M(a^{\infty})$＝秩 $D_A=W_H(A^N)$，$W_H(A^N)$ 表示 A^N 的汉明重量。类似地，可证明，$L(a^{\infty})=W_H(a^N)$。

Blahut 定理表明，一个周期半无限序列的线性复杂度等于它的一个周期的 DFT 的汉明重量。Blahut 定理在编码和密码的研究中起着重要的作用。

值得注意的是，在有限域上，不是对所有的 N 存在离散傅里叶变换，因为不是每一个数都可以作为元素的阶。但是，对大多数的目的来说，这已经足够了。如果 m 是 q^m-1 能够被 N 除尽的最小整数，那么在 F_q 上就有长为 N 的有限域离散傅里叶变换，其分量在 F_{q^m} 上。不幸的是，对于某些值，虽然变换是存在的，但是，它存在于很大的扩域中，因而不一定实用。

例 10.3.1 设 $a^{\infty}=0111010\cdots$ 是 F_2 上的一个周期为 7 的序列，α 是域 F_{2^3} 的一个本原元，即 α 的阶为 7。$a^7=0111010\cdots$ 的 DFT 为 $A^7=0\alpha\alpha^2 0\alpha^4 00\cdots$，由 Blahut 定理知，$L(a^{\infty})=W_H(A^7)=3$。反之，$L(A^{\infty})=W_H(a^7)=4$。

10.3.2 有限域上的其他频谱技术

类似于布尔函数和 m 值逻辑函数，下面来讨论有限域上的逻辑函数的两种谱。

设 $q=p^m$，其中 p 为素数。有限域 F_q 为素域 F_p 的扩域，由有限域的知识可知，F_q 是素域 F_p 的单扩张，即存在素域 F_p 上的某个代数次数为 m 的极小多项式 $u(x)$ 的根 α，使得 $F_q=F_p(\alpha)$，且存在唯一一个含 p^n 个元素的有限域，易知 $1,\alpha,\alpha^2,\cdots,\alpha^{m-1}$是 F_q 在 F_p 上的一组基。记 $F_q^*=F_q\backslash\{0\}$，则 F_q^* 是一个 $q-1$ 阶的乘法循环群，可取上述 α 为 F_q^* 的生成元。记 F_q^n 为 n 个 F_q 的笛卡儿积，又记 $W_H(w)$为 w 的不为零的分量的个数，即 w 的汉明重量，p 次本原单位根 $u=e^{\frac{2\pi i}{p}}$，$w=(w_1,w_2,\cdots,w_n)\in F_q^n$，$x=(x_1,x_2,\cdots,x_n)\in F_q^n$，$w\cdot x=w_1x_1+w_2x_2+\cdots+w_nx_n$，且运算均在有限域 F_q 上进行。

称 $F_q^n\to F_q$ 的任一映射 $f(x)$为 F_q^n 上的 n **元 q 值逻辑函数**，简称 q 值逻辑函数。

定义 10.3.1　设 $f(x)$，$x\in F_q^n$ 为 n 元 q 值逻辑函数，则称

$$S_f(w)=\frac{1}{q^n}\sum_{x\in F_q^n}\mathrm{tr}(f(x))\cdot u^{-\mathrm{tr}(w\cdot x)},\quad w\in F_q^n \tag{10.71}$$

和

$$S_{(f)}(w)=\frac{1}{q^n}\sum_{x\in F_q^n}u^{\mathrm{tr}(f(x))-\mathrm{tr}(w\cdot x)},\quad w\in F_q^n \tag{10.72}$$

分别为 $f(x)$的 Chrestenson 线性谱和 Chrestenson 循环谱。其中 $\mathrm{tr}(\cdot)$表示迹函数，即对任意的 $\beta\in F_q$，β 在 F_p 上的迹函数定义为

$$\mathrm{tr}(\beta)=\beta+\beta^p+\cdots+\beta^{p^{m-1}}$$

由迹函数的性质易证，$\{u^{\mathrm{tr}(w\cdot x)}\}_{w\in F_q^n}$是一个正交组。这样可得以下反演公式：

定理 10.3.2　设 $f(x)$，$x\in F_q^n$ 为 n 元 q 值逻辑函数，则

$$\mathrm{tr}(f(x))=\sum_{w\in F_q^n}S_f(w)u^{\mathrm{tr}(w\cdot x)},\quad x\in F_q^n \tag{10.73}$$

$$u^{\mathrm{tr}(f(x))}=\sum_{w\in F_q^n}S_{(f)}(w)u^{\mathrm{tr}(w\cdot x)},\quad x\in F_q^n \tag{10.74}$$

证明：由有限域上 Chrestenson 谱易得。

下面定理给出了两种谱之间的关系。

定理 10.3.3　设 $f(x)$，$x\in F_q^n$ 为 n 元 q 值逻辑函数，则

$$S_f(w)=\frac{1}{q}\sum_{i,k\in F_q}\mathrm{tr}(i)u^{-\mathrm{tr}(ki)}S_{(kf)}(w) \tag{10.75}$$

$$S_{(f)}(w)=\sum_{k\in F_q}S_{f+k}(w)u^{-\mathrm{tr}(k)}\Big/\sum_{i\in F_q}\mathrm{tr}(i)u^{-\mathrm{tr}(i)} \tag{10.76}$$

证明：记 $A_i(w)=\sum_{j\in F_q}u^{\mathrm{tr}(j)}P\{f(X)=i,w^*\cdot x=j\}$，其中 w^* 表示 w 的加法逆元。易知

$$S_{(0)}(w)=\sum_{i\in F_q}A_i(w)=\begin{cases}0 & w\neq 0\\ 1 & w=0\end{cases}$$

$$S_f(w)=\sum_{i\in F_q}\mathrm{tr}(i)A_i(w)$$

$$S_{(f)}(w)=\sum_{i\in F_q}u^{\mathrm{tr}(i)}A_i(w)$$

$$\begin{aligned}\sum_{i,k\in F_q}\mathrm{tr}(i)u^{-\mathrm{tr}(ki)}S_{(kf)}(w)&=\sum_{i\in F_q}\mathrm{tr}(i)\Big[\sum_{k\in F_q\setminus\{0\}}u^{-\mathrm{tr}(ki)}S_{(kf)}(w)+S_{(0)}(w)\Big]\\&=\sum_{i\in F_q}\mathrm{tr}(i)\sum_{k\in F_q\setminus\{0\}}u^{-\mathrm{tr}(ki)}\sum_{j\in F_q}u^{\mathrm{tr}(kj)}A_j(w)+\frac{(p-1)q}{2}S_{(0)}(w)\\&=\sum_{i\in F_q}\mathrm{tr}(i)\sum_{j\in F_q}A_j(w)\sum_{k\in F_q\setminus\{0\}}u^{\mathrm{tr}(k(j-i))}+\frac{(p-1)q}{2}S_{(0)}(w)\\&=\sum_{i\in F_q}\mathrm{tr}(i)\sum_{j\in F_q}A_j(w)\sum_{k\in F_q}u^{\mathrm{tr}(k(j-i))}\\&\quad-\sum_{i\in F_q}\mathrm{tr}(i)\sum_{j\in F_q}A_j(w)+\frac{(p-1)q}{2}S_{(0)}(w)\\&=q\sum_{i\in F_q}\mathrm{tr}(i)A_i(w)\\&=qS_f(w)\end{aligned}$$

$$\begin{aligned}\sum_{k\in F_q}S_{f+k}(w)u^{-\mathrm{tr}(k)}&=\sum_{k\in F_q}u^{-\mathrm{tr}(k)}\sum_{i\in F_q}\mathrm{tr}(i)A_{i-k}(w)\\&=\sum_{i\in F_q}\mathrm{tr}(i)u^{-\mathrm{tr}(i)}\sum_{k\in F_q}u^{\mathrm{tr}(i-k)}A_{i-k}(w)\\&=\sum_{i\in F_q}\mathrm{tr}(i)u^{-\mathrm{tr}(i)}S_{(f)}(w)\end{aligned}$$

而$\sum_{i\in F_q}\mathrm{tr}(i)u^{-\mathrm{tr}(i)}\neq 0$，故结论成立。

在此基础上有以下定理。

定理 10.3.4 设 $f(x),x\in F_q^n$ 为 n 元 q 值逻辑函数，则

$$S_f(w)=\frac{p-1}{2}S_{(0)}(w)+\sum_{k=1}^{p-1}\frac{1}{u^{-k}-1}S_{(kf)}(w)\tag{10.77}$$

$$S_{(f)}(w)=\frac{u^{-1}-1}{p}\sum_{k=0}^{p-1}S_{f+k}(w)u^{-\mathrm{tr}(k)}\tag{10.78}$$

证明：略。

下面给出一个应用。

定义 10.3.2 设 $f(x),x\in F_q^n$ 为 F_q^n 上的 n 元 q 值逻辑函数，$X_1,X_2,\cdots,X_n$ 为某概率空间$(\Omega,\mathcal{F},P)$上的 n 个相互独立且都具有均匀分布的随机变量。若对任意的 $1\leqslant i_1<\cdots<i_k\leqslant n$，$k\leqslant n$ 为取定的正整数，q 值随机变量 $f(X_1,X_2,\cdots,X_n)$与 k 维 q 值随机变量$(X_{i_1},X_{i_2},\cdots,X_{i_k})$都相互独立，则称 n 元 q 值逻辑函数 $f(x)$是 k 阶相关免疫的。又若 $P\{f(X)=a\}=\dfrac{1}{q}$，$a\in F_q$，则称 $f(x)$为平衡的。若 $f(x)$是 k 阶相关免疫的且平衡的，则称 $f(x)$是 k 阶弹性的。

定理 10.3.5 设 $f(x),x\in F_q^n$ 为 F_q^n 上的 n 元 q 值逻辑函数，则 $f(x)$是 k 阶相

关免疫的充要条件是对任意的 $a\in F_q\backslash\{0\}$ 及 $w\in F_q^n$：$1\leqslant W_H(w)\leqslant k$，有

$$S_{(af)}(w)=0 \tag{10.79}$$

例 10.3.2　取 $F_q=F_4=\{0,1,\alpha,\alpha^2\}$，其中 $\alpha^2+\alpha+1=0$。令

$$f(x_1,x_2,\cdots,x_n)=u\cdot x$$

则

$$S_{(f)}(w)=\frac{1}{4^n}\sum_{x\in F_4^n}(-1)^{\mathrm{tr}(u\cdot x)-\mathrm{tr}(w\cdot x)}=\frac{1}{4^n}\sum_{x\in F_4^n}(-1)^{\mathrm{tr}((u-w)\cdot x)}$$
$$=\begin{cases}1 & w=u\\ 0 & w\neq u\end{cases}$$

而且对任意的 $a\in F_q\backslash\{0\}$，有

$$S_{(af)}(w)=\frac{1}{4^n}\sum_{x\in F_4^n}(-1)^{\mathrm{tr}(au\cdot x)-\mathrm{tr}(w\cdot x)}=\frac{1}{4^n}\sum_{x\in F_4^n}(-1)^{\mathrm{tr}((au-w)\cdot x)}$$
$$=\begin{cases}1 & w=au\\ 0 & w\neq au\end{cases}$$

我们知道，在有限域上若 $xy=0$ 且 $y\neq0$，则 $x=0$。从而若 $W_H(u)=t$，则 $W_H(au)=t$，即 $f(x_1,x_2,\cdots,x_n)$是 $t-1$ 阶相关免疫逻辑函数。其中 $t=W_H(u)$。由计算过程可知，q 可以是任意的。

10.4　注记

本章重点介绍了一些在信息安全研究中常用的频谱方法与技术，同时用一些例子进行了说明。其目的是为了满足信息安全领域中的基本应用而选材的，感兴趣的读者可参阅文献[2]～[5]中的相关章节及其引用的参考文献。

参考文献

[1]　Ding C S, Xiao G Z, Shan W J. The Stability Theory of Stream Ciphers. Springer-Verlag, 1991

[2]　丁存生，肖国镇. 流密码学及其应用. 北京：国防科学出版社，1994

[3]　李世取，曾本胜，廉玉忠等. 密码学中的逻辑函数. 北京：北京中软电子出版社，2003

[4]　冯登国. 频谱理论及其在密码学中的应用. 北京：科学出版社，2000

[5]　温巧燕，钮心忻，杨义先. 现代密码学中的布尔函数[M]. 北京：科学出版社，2000

[6]　赵亚群. 多输出 m 值逻辑函数若干密码学性质的谱特征[R]. 中国科学院研究生院信息安全国家重点实验室博士后研究工作报告，2004

第 11 章　纠错码方法与技术

纠错码为避免数据传输过程中发生错误提供了一种不仅能检错而且能纠错的数学方法。现代纠错码的研究始于 Claude Shannon 发表于 1948 年的著名论文“通信中的数学理论”。Shannon 当时工作于贝尔实验室，他主要研究怎样解决电话通信中出现的问题，尤其是怎样纠正数据在电话线中传输时产生的错误。不仅如此，Shannon 于 1949 年还发表了著名的论文“保密通信的信息理论”，将密码学的研究纳入了科学的轨道，是对称密码学的奠基性文献。由此也不难推出，纠错码与密码学有着必然的联系。事实也证明的确如此。本章的重点是介绍一些在信息安全研究中常用的纠错码方法与技术，包括纠错码的基本概念、线性码和循环码的基本性质、一些好码的结构特征、一些典型的译码方法及典型应用实例。

11.1　基本概念

本节首先来定义纠错码中需要用到的一些基本概念。

11.1.1　码的定义和示例

定义 11.1.1　设 A 是一个有限集合，n 是一个正整数，一个长为 n 的码 C 是 $A^n=\{(a_1,a_2,\cdots,a_n)\mid a_i\in A,i=1,2,\cdots,n\}$ 的一个子集合。此时，则称 A^n 为码空间，A^n 中的元素称为字，C 中的元素称为码字。

特别地，当 $A=\{0,1\}$ 时，则称 C 是一个长为 n 的二元码。

假设 Alice 想要发送消息给 Bob，则将会用定义 11.1.1 中所定义的码字构成的串。每个码字代表一部分信息，而这个信息并不一定是一个英文单词。例如，一个码字可能代表英文中的一个字母，也可能代表 $A^m(m<n)$ 中的一个元素。码空间 A^n 是在考虑出错的情况下 Bob 可能收到的所有字的集合，码 C 是 Alice 可能发送的所有字的集合。

例 11.1.1　设 $A=\{0,1\}$，$n=3$，令 $C=A^n$，则有

$$A^n=\{000,001,010,011,100,101,110,111\}=C$$

当 $C=A^n$ 时，Bob 不能用这个编码方案检错，这是因为所有可能接收到的字均为码字，即均为 Alice 可能发送的字，所以 Bob 不能判断是否有错误发生。

例 11.1.2　设 $A=\{0,1\}$，$n=4$，则有

$$A^n=\{0000,0001,0010,0011,0100,0101,0110,0111\\ 1000,1001,1010,1011,1100,1101,1110,1111\}$$

令

$$C=\{0000,0011,0101,0110,1001,1010,1100,1111\}$$

例 11.1.2 中，$C \neq A^n$，即 Bob 可能收到的字集合大于 Alice 可能发送的字集合，所以 Bob 可能接收到一个 Alice 不可能发送的字。例如，Bob 可能接收到 0001，由于它不是 C 中的元素，所以他知道它不是 Alice 想要发送给他的，必然有一个错误发生。

定义 11.1.2　设 n 是一个正整数，$A=\{0,1\}$，则 A^n 中有 2^n 个元素。定义一个长为 n 的二元重复码 C 是只包含两个元素的集合，即只包含全 0 字符串和全 1 字符串。

例 11.1.3　设 $A=\{0,1\}$，$n=5$，则有

$$A^n=\{00000,00001,00010,00011,00100,00101,00110,00111,\\01000,01001,01010,01011,01100,01101,01110,01111,\\10000,10001,10010,10011,10100,10101,10110,10111,\\11000,11001,11010,11011,11100,11101,11110,11111\}$$

$$C=\{00000,11111\}$$

例 11.1.3 中的码能够纠正 2 个错。Bob 只需要在 5 比特中遵循少数服从多数的原则就可以了(这个原则也称**择多原则**)。例如，假设 Alice 发送给 Bob 的字符串为 11111，而他接收到的为 01011。他就会发现出了错，因为 Alice 只可能发送 2 个字：00000、11111。Bob 注意到 01011 中有 3 个 1，2 个 0，Alice 很有可能想要发送的是 11111。只要发生的错误不超过 2 个，这种纠错方法都会奏效。这种码的不足之处在于它极大地减少了 Alice 可能发送的信息量。由于只有两个码字，Alice 只能对 2 种信息片段进行编码。在 Alice 和 Bob 需要纠错能力较强而需要发送的信息量不大时，他们可以采用这种码。

定义 11.1.3　设 n 是一个正整数，$A=\{0,1\}$。定义一个长为 n 的二元奇偶校验码 C 是 A^n 中包含偶数个 1 的字构成的集合。

例 11.1.4　设 $A=\{0,1\}$，$n=5$，则有

$$A^n=\{00000,00001,00010,00011,00100,00101,00110,00111,\\01000,01001,01010,01011,01100,01101,01110,01111,\\10000,10001,10010,10011,10100,10101,10110,10111,\\11000,11001,11010,11011,11100,11101,11110,11111\}$$

$$C=\{00000,00011,00101,00110,01001,01010,01100,01111,\\10001,10010,10100,10111,11000,11011,11101,11110\}$$

这个奇偶校验码能够检测 1 位错误而不能纠正错误。这个码的优点在于 Alice 能够对很多的信息进行编码。它的不足之处在于它不能纠错。

11.1.2　Hamming 距离和码的极小距离

定义 11.1.4　设 $\boldsymbol{x},\boldsymbol{y}\in A^n$，定义 $\boldsymbol{x},\boldsymbol{y}$ 之间的 Hamming 距离 $d_{\mathrm{H}}(\boldsymbol{x},\boldsymbol{y})$ 为 $\boldsymbol{x}$、$\boldsymbol{y}$ 之间不同的比特位的个数。特别地，当 $\boldsymbol{y}=\boldsymbol{0}$ 时，称 $d_{\mathrm{H}}(\boldsymbol{x},\boldsymbol{0})$ 为 $\boldsymbol{x}$ 的 Hamming 重量或简称为重量，简记为 $W_{\mathrm{H}}(\boldsymbol{x})$。

例 11.1.5　设 $n=6$，$A=\{0,1\}$，$\boldsymbol{x}=(110100)$，$\boldsymbol{y}=(101010)$，则 $d_{\mathrm{H}}(\boldsymbol{x},\boldsymbol{y})=4$。

这是因为 $\boldsymbol{x},\boldsymbol{y}$ 在第二、三、四、五位共 4 位上不同。

例 11.1.6 设 $n=5,A=\{0,1,2,3,4\},\boldsymbol{x}=(34201),\boldsymbol{y}=(30210)$，则 $d_{\mathrm{H}}(\boldsymbol{x},\boldsymbol{y})=3$。这是因为 $\boldsymbol{x}$、$\boldsymbol{y}$ 在第二、四、五位共 3 位上不同。

关于 Hamming 距离，有以下事实。

定理 11.1.1 对任意的 $\boldsymbol{x},\boldsymbol{y},\boldsymbol{z}\in A^n$，Hamming 距离具有通常距离函数所具有的一些特性：

(1)（非负性）$d_{\mathrm{H}}(\boldsymbol{x},\boldsymbol{y})\geqslant 0$；

(2)（自反性）$d_{\mathrm{H}}(\boldsymbol{x},\boldsymbol{y})=0$ 当且仅当 $\boldsymbol{x}=\boldsymbol{y}$；

(3)（对称性）$d_{\mathrm{H}}(\boldsymbol{x},\boldsymbol{y})=d_{\mathrm{H}}(\boldsymbol{y},\boldsymbol{x})$；

(4)（三角不等式）$d_{\mathrm{H}}(\boldsymbol{x},\boldsymbol{y})\leqslant d_{\mathrm{H}}(\boldsymbol{x},\boldsymbol{z})+d_{\mathrm{H}}(\boldsymbol{z},\boldsymbol{y})$。

证明：前 3 个性质是显然的，这里只证第四个性质。设

$$\boldsymbol{x}=(x_1,x_2,\cdots,x_n)$$
$$\boldsymbol{y}=(y_1,y_2,\cdots,y_n)$$
$$\boldsymbol{z}=(z_1,z_2,\cdots,z_n)$$

显然当 $x_i\neq y_i(i=1,2,\cdots,n)$时，一定有 $x_i\neq z_i$ 或 $y_i\neq z_i$。
因此，由 Hamming 距离的定义可知，$d_{\mathrm{H}}(\boldsymbol{x},\boldsymbol{y})\leqslant d_{\mathrm{H}}(\boldsymbol{x},\boldsymbol{z})+d_{\mathrm{H}}(\boldsymbol{z},\boldsymbol{y})$。

定理 11.1.1 表明，d_{H} 是 A^n 中的一个度量。

寻找“好”码是纠错码研究中的一个核心问题。但是，什么是“好”码呢？首先要有足够大的码字集合，这样就可以对很多的信息进行编码，其次能够发现并纠正尽量多的错误。给定一个码 C，怎样来决定它能检测多少错误？能纠正多少错误？假设有一个码 C，它的所有码字之间的 Hamming 距离至少为 3。假设 Alice 向 Bob 发送一个字，我们知道这其中至多有一个错误，则接收到的字 $\boldsymbol{r}$ 与本来要发送的码字 $\boldsymbol{x}$ 之间的 Hamming 距离为 1 或者为 0。有没有这样一种可能：存在另外一个码字与 $\boldsymbol{r}$ 的距离不大于 1？假设有，即存在 $\boldsymbol{y}\in C,\boldsymbol{y}\neq\boldsymbol{x}$，满足 $d_{\mathrm{H}}(\boldsymbol{r},\boldsymbol{y})\leqslant 1$。由三角不等式，有 $d_{\mathrm{H}}(\boldsymbol{x},\boldsymbol{y})\leqslant d_{\mathrm{H}}(\boldsymbol{x},\boldsymbol{r})+d_{\mathrm{H}}(\boldsymbol{r},\boldsymbol{y})\leqslant 2$。所以，存在两个码字，它们之间的距离小于 3，而这与之前关于所有码字之间的距离不小于 3 的假定矛盾。这说明 $\boldsymbol{x}$ 是与 $\boldsymbol{r}$ 唯一最近的码字，所以 Bob 能够很有信心地把 $\boldsymbol{r}$ 译码为 $\boldsymbol{x}$。只要发生的错误不超过 1，Bob 就能够正确地纠错。注意到关于所有码字之间的距离不小于 3 的假定是很重要的，把这一点用数学的语言描述，有下面的定义。

定义 11.1.5 设 $C\subseteq A^n$ 是一个码。定义该码的极小距离为

$$\min_{\boldsymbol{x},\boldsymbol{y}\in C,\boldsymbol{x}\neq\boldsymbol{y}} d_{\mathrm{H}}(\boldsymbol{x},\boldsymbol{y})$$

记一个码的极小距离为 $d(C)$，有时也简记为 d。

为了找到一个码的极小距离，只要计算所有码字之间的 Hamming 距离即可，得到的最小值即为这个码的极小距离。

例 11.1.7 设 C 是长为 4 的二元重复码，则有 $C=\{(0000),(1111)\}$，则 C 的极小距离为 4。事实上，长为 n 的二元重复码的极小距离为 n。

例 11.1.8 设 C 为长为 5 的二元奇偶校验码，可以直接验证 C 的极小距离为 2。事实上，长为 n 的二元奇偶校验码的极小距离为 2。

码 C 的极小距离可用来度量该码所能检测和纠正错误的能力。

首先要说明何谓一个码所能检测的错误的数目。当 Bob 接收到 Alice 发送过来的字 $\boldsymbol{r}$，他检验 $\boldsymbol{r}$ 是不是 C 中的元素。如果是，他认为没有错误，否则，他认为至少有一个错误存在于 Alice 的发送过程中。如果无论何时发生不多于 e 个的错误，无论 Alice 发送的码字如何，Bob 都能够准确地说出是否有错误发生，就说码 C 能检测 e 个错误。

其次，要说明何谓一个码所能纠正的错误数目。当 Bob 接收到一个码字，他把它译码为 $\boldsymbol{x}$，使得 $d_{\mathrm{H}}(\boldsymbol{x},\boldsymbol{r})$ 足够小。如果无论何时发生不多于 e 个的错误，无论 Alice 发送的码字如何，Bob 都能够保证准确地纠正这些错误。要记住的是 Bob 能准确地把接收到的字 $\boldsymbol{r}$ 译码的前提是 Alice 发送过来的码字是与 $\boldsymbol{r}$ 唯一最近的码字。

现在证明如果知道一个码字的极小距离，就能知道他能够检测多少错误，纠正多少错误。

定理 11.1.2　设 $C\subseteq A^n$ 是一个极小距离为 d 的码，则 C 能够检测 $d-1$ 个错误。

证明：假设 Alice 发送给 Bob 的码字为 $\boldsymbol{c}\in C$。假定发生的错误不多于 $d-1$ 个，将证明 Bob 能够正确地指出是否有错误发生。设 Bob 接收到的字为 $\boldsymbol{r}\in A^n$，假设没有错误发生，则 $\boldsymbol{r}=\boldsymbol{c}\in C$，所以 Bob 正确地认为没有错误发生；另一方面，假设在传输中有错误发生，而且错误的个数不多于 $d-1$，则 $1\leqslant d_{\mathrm{H}}(\boldsymbol{c},\boldsymbol{r})\leqslant d-1$。由于码 C 的极小距离为 d，故 $\boldsymbol{r}\notin C$。所以 Bob 正确地认为有错误发生，即说明 C 能够检测 $d-1$ 个错误。

定理 11.1.3　设 $C\subseteq A^n$ 是一个极小距离为 d 的码，则 C 能够纠正 $e=\left\lfloor\frac{d-1}{2}\right\rfloor$个错误。其中$\lfloor l\rfloor$表示不大于 l 的最大的正整数。

证明：设 Alice 发送给 Bob 的码字为 $\boldsymbol{c}\in C$，Bob 接收到的字为 $\boldsymbol{r}\in A^n$。假定发生的错误不多于 $e=\left\lfloor\frac{d-1}{2}\right\rfloor$个，将证明 Bob 能够正确地纠正错误。注意到 $d_{\mathrm{H}}(\boldsymbol{c},\boldsymbol{r})\leqslant e$，想要说明 $\boldsymbol{c}$ 为 C 中与 $\boldsymbol{r}$ 最近的码字。反之假设存在 $\boldsymbol{y}\in C$，$\boldsymbol{y}\neq\boldsymbol{c}$，满足 $d_{\mathrm{H}}(\boldsymbol{r},\boldsymbol{y})\leqslant e$。由三角不等式，则有

$$d_{\mathrm{H}}(\boldsymbol{c},\boldsymbol{y})\leqslant d_{\mathrm{H}}(\boldsymbol{c},\boldsymbol{r})+d_{\mathrm{H}}(\boldsymbol{r},\boldsymbol{y})\leqslant e+e=2e=2\left\lfloor\frac{d-1}{2}\right\rfloor\leqslant d-1$$

所以，存在两个码字，它们之间的距离小于 d，而这与之前关于码 C 的极小距离为 d 的假定矛盾。即说明了 C 能够纠正 e 个错误。

例 11.1.9　长为 n 的二元重复码的极小距离为 n，所以它能检测 $n-1$ 个错误，纠正$\left\lfloor\frac{n-1}{2}\right\rfloor$个错误。

例 11.1.10　长为 n 的二元奇偶校验码的极小距离为 2，所以它只能检测 1 个错误，而不能纠错。

上述定理表明，码的极小距离越大，它的检错能力和纠错能力也就相应的越大。因此，在纠错码中总是要求码具有较大的极小距离。

定义 11.1.6 设 $C \subseteq A^n$ 是一个码，$\boldsymbol{x} \in A^n$。定义 $\boldsymbol{x}$ 与码 C 的 Hamming 距离为

$$d_{\mathrm{H}}(\boldsymbol{x}, C) = \min_{\boldsymbol{c} \in C} d_{\mathrm{H}}(\boldsymbol{x}, \boldsymbol{c})$$

定义该码的覆盖半径为

$$\max_{\boldsymbol{x} \in A^n} \min_{\boldsymbol{c} \in C} \ d_{\mathrm{H}}(\boldsymbol{x}, \boldsymbol{c})$$

记一个码的覆盖半径为 $\rho(C)$，有时也简记为 ρ。

11.2 线性码和循环码

11.2.1 线性码的定义和基本性质

令 $A=F$，其中 F 是一个域，此时 $A^n=F^n$ 是 F 上的一个 n 维向量空间。

定义 11.2.1 设 F_q 是一个含 q 个元素的有限域。令 $C \subseteq F_q^n$ 为一个码。我们说一个码 C 是线性的，如果

（1）C 非空；

（2）对任意的 $\boldsymbol{x}, \boldsymbol{y} \in C$，$\boldsymbol{x}+\boldsymbol{y} \in C$；

（3）对每一个 $\alpha \in F_q$，每一个 $\boldsymbol{x} \in C$，都有 $\alpha \boldsymbol{x} \in C$。

换句话说，一个码 C 是线性的当且仅当它是非空的而且在加法和数乘下封闭。说一个码 C 是线性的，也就是说 C 是向量空间 F_q^n 的一个子空间。

容易验证，前面介绍的二元重复码和二元奇偶校验码都是域 $F_2 = \ _2$ 上的线性码。

例 11.2.1 设 $A=F_2$，$n=4$，$C=\{(1100),(1110)\}$，则 C 不是线性码，因为它在加法下不封闭，特别地，$(1100)+(1110)=(0010) \notin C$。

显然，如果 C 是线性的，则每一位都为零的字必然在 C 中。这是因为，若令 $\boldsymbol{0}$ 表示每一位都为零的字，注意到 $0 \in F_q$。由于 C 是线性的，则 C 必非空，所以存在 $\boldsymbol{x} \in C$。但是 $0\boldsymbol{x}=\boldsymbol{0}$，所以每一位都为零的字在 C 中。同理可证，若 $\boldsymbol{x} \in C$，则 $-\boldsymbol{x} \in C$。

令 $\boldsymbol{x} \in F_q^n$，由定义 11.1.6 可知，$\boldsymbol{x}$ 的 Hamming 重量或重量定义为 $d_{\mathrm{H}}(\boldsymbol{x}, \boldsymbol{0})$，即为 $\boldsymbol{x}$ 的非零分量的个数，简记为 $W_{\mathrm{H}}(\boldsymbol{x})$。

例 11.2.2 若 $\boldsymbol{x}=(1001110) \in F_2^7$，则 $W_{\mathrm{H}}(\boldsymbol{x})=4$。若 $\boldsymbol{x}=(41060) \in F_7^5 = \ _7^5$，则 $W_{\mathrm{H}}(\boldsymbol{x})=3$。

定义 11.2.2 设 $C \subseteq F_q^n$ 是一个码，定义码 C 的极小重量 $W_{\min}(C)$ 为

$$W_{\min}(C) = \min_{\boldsymbol{x} \in C, \boldsymbol{x} \neq 0} \{W_{\mathrm{H}}(\boldsymbol{x})\}$$

所以，为了求码 C 的极小重量，只要计算 C 中非零码字的重量就可以了。其中得到的最小值就是码 C 的极小重量。

例 11.2.3 长为 n 的二元重复码的极小重量为 n，长为 n 的二元奇偶校验码的极小重量为 2。

定理 11.2.1 设 $C \subseteq F_q^n$ 是一个线性码，则 $W_{\min}(C)=d_{\min}(C)$。

证明： 设 $\boldsymbol{x} \in C$，$\boldsymbol{x} \neq \boldsymbol{0}$，则 $W_{\mathrm{H}}(\boldsymbol{x})=d_{\mathrm{H}}(\boldsymbol{x}, \boldsymbol{0})$。

因此

$$W_{\min}(C)=\min_{x\in C,x\neq 0}\{W_{\mathrm{H}}(\boldsymbol{x})\}=\min_{x\in C,x\neq 0}\{d_{\mathrm{H}}(\boldsymbol{x},\boldsymbol{0})\}$$
$$\geqslant\min_{x,y\in C,x\neq y}d_{\mathrm{H}}(\boldsymbol{x},\boldsymbol{y})=d_{\min}(C)$$

另一方面，如果 $\boldsymbol{x}\neq\boldsymbol{y}$ 为 C 中的两个元素，则有

$$d_{\mathrm{H}}(\boldsymbol{x},\boldsymbol{y})=W_{\mathrm{H}}(\boldsymbol{x}-\boldsymbol{y})$$

由于 C 是线性的，所以 $\boldsymbol{x}-\boldsymbol{y}\in C$。注意到 $\boldsymbol{x}-\boldsymbol{y}\neq\boldsymbol{0}$，所以有

$$d_{\min}(C)=\min_{x,y\in C,x\neq y}d_{\mathrm{H}}(\boldsymbol{x},\boldsymbol{y})=\min_{x,y\in C,x\neq y}W_{\mathrm{H}}(\boldsymbol{x}-\boldsymbol{y})$$
$$\geqslant\min_{x\in C,x\neq 0}\{W_{\mathrm{H}}(\boldsymbol{x})\}=W_{\min}(C)$$

综上所述

$$d_{\min}(C)=W_{\min}(C)$$

值得注意的是，计算一个码的极小重量要比计算它的极小距离容易，所以这个定理使得码的极小距离的计算更容易。这一点是十分有用的，因为如果知道了一个码的极小距离，就能够决定它能够检测多少错误，纠正多少错误。

11.2.2　生成矩阵

定理 11.2.1 表明，线性码能够通过计算码的极小重量来计算它的极小距离。不仅如此，线性码还存在一个简明的表示——生成矩阵表示。

定义 11.2.3　设 $\boldsymbol{G}$ 是 F_q 上的一个 $k\times n$ 阶矩阵，令 $\boldsymbol{x}_i$ 为 $\boldsymbol{G}$ 的第 i 行。定义 $\boldsymbol{G}$ 的行空间(记为 RS($\boldsymbol{G}$))为

$$\mathrm{RS}(\boldsymbol{G})=\{\alpha_1\boldsymbol{x}_1+\alpha_2\boldsymbol{x}_2+\cdots+\alpha_k\boldsymbol{x}_k\mid\alpha_i\in F_q\}\subseteq F_q^n。$$

显然，RS($\boldsymbol{G}$)是由 $\boldsymbol{G}$ 的行向量张成的子空间。

例 11.2.4　若 $F_q=F_2$，$\boldsymbol{G}=\begin{pmatrix}1&0&0&1\\0&1&0&1\\0&0&1&1\end{pmatrix}$，则

$$\mathrm{RS}(\boldsymbol{G})=\{(0000),(1001),(0101),(0011),(1100),(1010),(0110),(1111)\}$$

这是一个长为 4 的奇偶校验码。

例 11.2.5　若 $F_q=F_2$，$\boldsymbol{G}=(1\quad 1\quad 1\quad 1\quad 1)$，则

$$\mathrm{RS}(\boldsymbol{G})=\{(00000),(11111)\}$$

这是一个长为 5 的二元重复码。

若 $C\subseteq F_q^n$ 是一个线性码，并且包含多于一个的元素，则存在 F_q 上的一个 $k\times n$ 阶矩阵 $\boldsymbol{G}$，其中 $\boldsymbol{G}$ 的行向量是线性无关的(也称线性独立的)，且 $C=\mathrm{RS}(\boldsymbol{G})$。为了找到这样一个 $\boldsymbol{G}$，只要找到 C 的一组基，并把这一组基元素作为矩阵 $\boldsymbol{G}$ 的行向量。

定义 11.2.4　设 $C\subseteq F_q^n$ 是一个线性码，令 $\boldsymbol{G}$ 为 F_q 上的一个 $k\times n$ 阶矩阵，且 $\boldsymbol{G}$ 的行向量是线性独立的，$C=\mathrm{RS}(\boldsymbol{G})$，则 $\boldsymbol{G}$ 叫做 C 的生成矩阵，C 叫做参数为$[n,k]$的线性码，k 叫做 C 的维数。若 $C=\{\boldsymbol{0}\}$，就说 C 是一个 0 维的线性码。

如果知道一个线性码的生成矩阵，就能够通过计算这个矩阵的行向量空间来得到 C 的所有元素。也即是说，生成矩阵提供了一个描述线性码的简明方法。

值得一提的是，线性码的生成矩阵并不是唯一的，每一个线性码可以有多个生成

矩阵。例如,如果把任意一个生成矩阵的其中两行交换就可得到另一个矩阵,这个矩阵也是生成矩阵。由线性代数的知识可以知道,一个向量空间可以有多组基。由于把基中的元素作为 $\boldsymbol{G}$ 的行向量,这说明对一个线性码而言,可以找到多组基。但在纠错码中通常选择具有特别好的性质的生成矩阵。

11.2.3 对偶码和校验矩阵

定义 11.2.5 设 $C\subseteq F_q^n$ 是一个码,C 的对偶码(记为 $C^{\perp}$)定义为

$$C^{\perp}=\{\boldsymbol{x}\in F_q^n \mid \boldsymbol{x}\cdot\boldsymbol{c}=0\text{ 对任意的 }\boldsymbol{c}\in C\}\subseteq F_q^n$$

其中 $\boldsymbol{x}\cdot\boldsymbol{c}$ 表示域 F 上的点积。当 $\boldsymbol{x}\cdot\boldsymbol{c}=0$ 时,也称 $\boldsymbol{x}$ 和 $\boldsymbol{c}$ 正交。

例 11.2.6 设 C 是长为 4 的二元奇偶校验码。所以,

$$C=\{(0000),(0011),(0101),(1001),(0110),(1100),(1111)\}$$

为了找到 $C^{\perp}$ 中的元素,对 F_2^4 中的所有元素进行测试,看哪些元素和 C 中所有元素的点积为 0。这里的所有运算都是在 F_2 中进行的。所以,例如,当测试(0111)是否在 $C^{\perp}$ 中时,注意到在 F_2 中 $(0111)\cdot(1111)=0+1+1+1=1$。由于存在一个码字(1111)与(0111)的点积不为 0,所以(0111)不在 $C^{\perp}$ 中。用这种方法跑遍 F_2^4 中的所有元素,可以验证

$$C^{\perp}=\{(0000),(1111)\}$$

这是一个长为 4 的二元重复码。当计算长为 4 的二元重复码的对偶码时,发现其正是长为 4 的二元奇偶校验码。也即是说,在这种情况下,$(C^{\perp})^{\perp}=C$。

值得注意的是,$(C^{\perp})^{\perp}=C$ 不是对所有的码都成立。

例 11.2.7 设 $C=\{(001),(010),(110)\}$,则 $C^{\perp}=\{(000)\}$,

$$(C^{\perp})^{\perp}=\{(000),(001),(010),(011),(100),(101),(110),(111)\}\neq C$$

事实上,对任意的码 C,无论其是否为线性的,则 $C^{\perp}$ 都是线性的。可由定义直接推出。

根据对偶码的定义,确定一个 F_q^n 中的元素是否为一个码的对偶码中的元素,需要计算这个元素与这个码的所有元素的点积。现在证明一个定理,这个定理给出一个对线性码而言较容易的方法。事实上,使用一个生成矩阵来判断一个元素是否在一个码的对偶码中。用 $\boldsymbol{x}^{\mathrm{T}}$ 来表示 $\boldsymbol{x}\in F_q^n$ 的转置。

定理 11.2.2 设 $C\subseteq F_q^n$ 是一个线性码,$\boldsymbol{G}$ 是 C 的生成矩阵。一个字 $\boldsymbol{x}\in F_q^n$ 在 $C^{\perp}$ 中当且仅当 $\boldsymbol{G}\boldsymbol{x}^{\mathrm{T}}=\boldsymbol{0}$。

证明:令 $\boldsymbol{c}_1,\boldsymbol{c}_2,\cdots,\boldsymbol{c}_k\in F_q^n$ 为 $\boldsymbol{G}$ 的行。现在假设 $\boldsymbol{x}\in C^{\perp}$,则 $\boldsymbol{x}\cdot\boldsymbol{c}=\boldsymbol{0}$ 对任意的 $\boldsymbol{c}\in C$。特别地,$\boldsymbol{x}\cdot\boldsymbol{c}_i=\boldsymbol{0}$ 对所有的 $i=1,2,\cdots,k$ 成立,这说明 $\boldsymbol{G}\boldsymbol{x}^{\mathrm{T}}=\boldsymbol{0}$。

反之,假设 $\boldsymbol{G}\boldsymbol{x}^{\mathrm{T}}=\boldsymbol{0}$ 对某一 $\boldsymbol{x}\in F_q^n$ 成立,则 $\boldsymbol{x}\cdot\boldsymbol{c}_i=0$ 对所有的 $i=1,2,\cdots,k$ 成立。对任意的 $\boldsymbol{c}\in C$,有

$$\boldsymbol{c}=\alpha_1\boldsymbol{c}_1+\alpha_2\boldsymbol{c}_2+\cdots+\alpha_k\boldsymbol{c}_k$$

对某些 $\alpha_i\in F_q$ 成立。这说明

$$\boldsymbol{c}\cdot\boldsymbol{x}=\alpha_1(\boldsymbol{c}_1\cdot\boldsymbol{x})+\alpha_2(\boldsymbol{c}_2\cdot\boldsymbol{x})+\cdots+\alpha_k(\boldsymbol{c}_k\cdot\boldsymbol{x})=0$$

所以,有 $\boldsymbol{x}\in C^{\perp}$,命题得证。

定理 11.2.3　若 $C \subseteq F_q^n$ 是一个参数为$[n,k]$的线性码，则 $C^{\perp}$ 是参数为$[n,n-k]$的线性码，并且$(C^{\perp})^{\perp}=C$。

证明：$C^{\perp}$ 是一个线性码这一事实可由定义直接推出。令 $\boldsymbol{G}$ 是 C 的一个生成矩阵，则 $\boldsymbol{G}$ 是一个秩为 k 的 $k\times n$ 阶矩阵。由定理 11.2.2 可知，码 C 的对偶码的维数等于 $\boldsymbol{G}$ 的补空间$\{\boldsymbol{x}\in F_q^n \mid \boldsymbol{G}\boldsymbol{x}^{\mathrm{T}}=\boldsymbol{0}\}$的维数。但由线性代数知识可知，一个矩阵与它的补空间的维数之和等于 n，所以 $\boldsymbol{G}$ 的补空间的维数等于 $n-k$，所以，$C^{\perp}$ 是参数为$[n,n-k]$的线性码。同理可证，$(C^{\perp})^{\perp}$ 的维数为 k。容易证明，对任意的码 C，都有 $C\subseteq(C^{\perp})^{\perp}$。所以，在这种情况下，有一个维数为 k 的子空间包含在另外一个维数为 k 的子空间中，因此，这两个子空间相等，即$(C^{\perp})^{\perp}=C$。

事实证明，上面的定理是很有用的。考虑一个线性码 C，由于知道 $C^{\perp}$ 是一个线性码，它必然有一个生成矩阵，就把这个矩阵称为 $\boldsymbol{H}$。由定理 11.2.2 可知，$\boldsymbol{x}\in(C^{\perp})^{\perp}$ 当且仅当 $\boldsymbol{H}\boldsymbol{x}^{\mathrm{T}}=\boldsymbol{0}$，但是由于知道$(C^{\perp})^{\perp}=C$，这也就是说，$\boldsymbol{x}\in C$ 当且仅当 $\boldsymbol{H}\boldsymbol{x}^{\mathrm{T}}=\boldsymbol{0}$。换句话说，矩阵 $\boldsymbol{H}$ 使得我们能够检验 F_q^n 中的元素是否在码中。用数学的语言，即为

$$C=\{\boldsymbol{x}\in F_q^n \mid \boldsymbol{H}\boldsymbol{x}^{\mathrm{T}}=\boldsymbol{0}\}$$

因此，有下面的定义。

定义 11.2.6　设 $C\subseteq F_q^n$ 是一个线性码，$C^{\perp}$ 的一个生成矩阵叫做 C 的一个校验矩阵。

现在通过校验矩阵定义一类著名的码——Hamming 码。二元和非二元 Hamming 码都存在，这里只介绍二元 Hamming 码。

定义 11.2.7　设 $\boldsymbol{H}$ 是一个 F_2 上的 $m\times(2^m-1)$阶矩阵，其列由 F_2 上的所有 2^m-1个非零 m 维列向量组成，以 $\boldsymbol{H}$ 为校验矩阵的二元 Hamming 码定义为

$$C=\{\boldsymbol{x}\in F_2^{2^m-1} \mid \boldsymbol{H}\boldsymbol{x}^{\mathrm{T}}=\boldsymbol{0}\}$$

显然，上述定义的二元 Hamming 码是一个参数为$[2^m-1,2^m-1-m]$的线性码。事实上，可以证明，这类码的极小距离 $d=3$。因为 $\boldsymbol{H}$ 的列向量非零且任何两列都不同，所以重量为 1 和 2 的字 $\boldsymbol{x}\in F_2^{2^m-1}$ 不能使 $\boldsymbol{H}\boldsymbol{x}^{\mathrm{T}}=\boldsymbol{0}$，因此这些字都不是码字。但存在重量为 3 的字 $\boldsymbol{x}\in F_2^{2^m-1}$ 使得 $\boldsymbol{H}\boldsymbol{x}^{\mathrm{T}}=\boldsymbol{0}$，所以由定理 11.2.1 可知，$d=3$。

例 11.2.8　参数为$[7,4]$的二元 Hamming 码的校验矩阵为

$$\boldsymbol{H}=\begin{pmatrix}1&1&1&0&1&0&0\\1&1&0&1&0&1&0\\0&1&1&1&0&0&1\end{pmatrix}$$

设 $\boldsymbol{x}=(x_1,x_2,x_3,x_4,x_5,x_6,x_7)$是 F_2^7 中的一个字，则由定义 11.2.7 可知，$\boldsymbol{x}$ 在 C 中当且仅当 $\boldsymbol{H}\boldsymbol{x}^{\mathrm{T}}=\boldsymbol{0}$。换句话说，$\boldsymbol{x}\in C$ 当且仅当下面 3 个式子成立：

$$(1,1,1,0,1,0,0)\cdot(x_1,x_2,x_3,x_4,x_5,x_6,x_7)=0;$$
$$(1,1,0,1,0,1,0)\cdot(x_1,x_2,x_3,x_4,x_5,x_6,x_7)=0;$$
$$(0,1,1,1,0,0,1)\cdot(x_1,x_2,x_3,x_4,x_5,x_6,x_7)=0。$$

其中所有的运算都是模 2 运算。但是，上面的等式成立当且仅当

$$x_5 = x_1 + x_2 + x_3$$
$$x_6 = x_1 + x_2 + x_4$$
$$x_7 = x_2 + x_3 + x_4$$

所以，若 $\boldsymbol{x} \in C$ 当且仅当它具有这样的形式：

$$(x_1, x_2, x_3, x_4, x_1 + x_2 + x_3, x_1 + x_2 + x_4, x_2 + x_3 + x_4)$$

但是可以把它写为

$$x_1(1000110) + x_2(0100111) + x_3(0010101) + x_4(0001011)$$

所以如果令

$$\boldsymbol{G} = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{pmatrix}$$

则 $C = \mathrm{RS}(\boldsymbol{G})$。容易验证 $\boldsymbol{G}$ 的行是线性独立的，所以 $\boldsymbol{G}$ 是 C 的一个生成矩阵。

11.2.4 Singleton 界和 MDS 码

下面的定理揭示了线性码的校验矩阵与其极小重量亦即极小距离之间的关系。

定理 11.2.4 设 $C \subseteq F_q^n$ 是一个参数为$[n,k]$的线性码，令 $u \in C$，$W_{\mathrm{H}}(u) = m$，C 的校验矩阵为$\boldsymbol{H}$，则 $\boldsymbol{H}$ 中有 m 列存在一个线性相关关系。反之，对于 $\boldsymbol{H}$ 的 m 列中任何一个线性相关关系，都对应一个 C 中重量不大于 m 的码字。

证明：因为 $\boldsymbol{u} \in C$，$\boldsymbol{H}$ 是 C 的校验矩阵，所以 $\boldsymbol{H}\boldsymbol{u}^{\mathrm{T}} = \boldsymbol{0}$，又 $W_{\mathrm{H}}(\boldsymbol{u}) = m$，去掉 $\boldsymbol{u}$ 的零分量，这正是 $\boldsymbol{H}$ 的 m 列的一个线性相关关系。因此，$\boldsymbol{H}$ 中有 m 列存在一个线性相关关系。反之，如果 $\boldsymbol{H}$ 的 m 列有一个线性相关关系，则存在 m 个不全为零的系数，使得这 m 列的线性组合等于零，现在定义一个一维行向量 $\boldsymbol{u}$：与这 m 个列对应的分量就取对应的这 m 个系数，其余分量取零。显然，$W_{\mathrm{H}}(\boldsymbol{u}) \leqslant m$，$\boldsymbol{H}\boldsymbol{u}^{\mathrm{T}} = \boldsymbol{0}$，所以，$\boldsymbol{u}$ 是一个 C 中重量不大于 m 的码字。

推论 11.2.1 设 $C \subseteq F_q^n$ 是一个线性码，C 的校验矩阵为 $\boldsymbol{H}$，则 C 的极小重量亦即极小距离为 d 当且仅当 $d = \max\{m \mid \boldsymbol{H}$ 的任意 $m-1$ 列都线性无关$\}$。

证明：由定理 11.2.4 可得。

定理 11.2.5（Singleton 界） 设 $C \subseteq F_q^n$ 是一个参数为$[n,k]$的线性码，则 C 的极小距离 $d \leqslant n-k+1$。

证明：设参数为$[n,k]$的线性码 C 的校验矩阵为 $\boldsymbol{H}$，则 $\boldsymbol{H}$ 是一个秩为 $n-k$ 的$(n-k) \times n$矩阵，因此，$\boldsymbol{H}$ 中的任意 $n-k+1$ 列都线性相关。由定理 11.2.4 可知，$d \leqslant n-k+1$。

定理 11.2.5 表明，一个线性码的极小距离不会"太大"，无论怎样努力，都不能够构造出一个参数为$[n,k]$的线性码，使得它的极小距离大于 $n-k+1$。由此可见，最好的期望就是构造使得极小距离等于 $n-k+1$ 的线性码，于是引出下面的定义。

定义 11.2.8 一个参数为$[n,k]$的线性码 C，若满足 $d_{\min}(C) = n-k+1$，则称该码为极大距离可分码，简称为 MDS 码。

事实证明，MDS 码是存在的，11.3.2 小节将要介绍的广义 Reed-Solomon 码就是 MDS 码。

11.2.5　循环码

循环码是一类非常重要的线性码，这类码自 1957 年 Prange 提出以来，有关码的结构、性质和编译码方法得到了极其快速和深入的研究，并且目前发现的大部分线性码都与其有着密切的联系。另外，由于这类码的代数结构清晰、性能较好、编译码电路简单和易于实现等特点，实际应用中使用的几乎都是这类码。

定义 11.2.9　设 F_q 是一个有限域。令 $C \subseteq F_q^n$ 为一个线性码。我们说一个线性码 C 是循环的，如果对任意的 $\mathbf{c}=(c_0,c_1,\cdots,c_{n-1})\in C$，$\mathbf{c}$ 的循环右移 $\mathbf{c}'=(c_{n-1},c_0,c_1,\cdots,c_{n-2})\in C$（因此，$\mathbf{c}$ 的任意循环移位都属于 C）。

换句话说，一个线性码 C 是循环的也就是说它是向量空间 F_q^n 的一个循环子空间。

例 11.2.9　例 11.2.8 中的参数为[7,4]的二元 Hamming 码就是一个循环码。

循环码的表示方法有很多，这里介绍 3 种常用的表示方法，即多项式表示、矩阵表示和根表示。

首先，讨论循环码的多项式表示。

定义 11.2.10　设 F 是一个域，多项式的集合 $F[x]$ 定义为

$$F[x] = \{r_0 + r_1x + r_2x^2 + \cdots + r_nx^n \mid r_i \in F, n \in N\}$$

集合 $F[x]$中有自然的加法和乘法，即通常意义上的加法和乘法，所有的运算都是在域 F 中。

例 11.2.10　设 $F=F_7=Z_7$，则 $F_7[x]$是所有系数在 F_7 中的多项式集合。此时，

$$(5+2x+3x^2+6x^3)+(3+2x+6x^2) = 1+4x+2x^2+6x^3$$

$$3(5+2x+3x^2+6x^3) = 1+6x+2x^2+4x^3$$

$$(5+x)(2+2x) = 3+5x+2x^2$$

定义 11.2.11　集合 $F[x]$及其定义在之上的加法和乘法被称为域 F 上的不定元为 x 的多项式环。

定义 11.2.12　设 k 是一个非负正整数，F 是一个域，则 $F[x]_k$ 表示 $F[x]$中所有次数小于 k 的多项式集合。按照惯例，零多项式的次数定义为 $-\infty$。

显然，$F[x]$是 F 上的无限维的向量空间，$F[x]_k$ 是 $F[x]$的维数为 k 的子空间。

设 F_q 是一个有限域。在 F_q^n 与 $F_q[x]_n$ 之间建立以下对应关系：对每个 $\mathbf{c}=(c_0,c_1,\cdots,c_{n-1})\in F_q^n$，令 $\boldsymbol{\Psi}(\mathbf{c})=c_0+c_1x+\cdots+c_{n-1}x^{n-1}\in F_q[x]_n$，则 $\boldsymbol{\Psi}: F_q^n \to F_q[x]_n$ 是一个一一对应关系。

定义 11.2.13　设 $C\subseteq F_q^n$ 是一个线性码，$\mathbf{c}=(c_0,c_1,\cdots,c_{n-1})\in C$，码字 $\mathbf{c}$ 的码多项式定义为

$$c(x) = c_0 + c_1x + \cdots + c_{n-1}x^{n-1}$$

由定义 11.2.13 可知，码多项式的系数就是码字各分量的值，x^i 的次数 i 代表该分量所在的位置。

令 $C(x)=\{c(x)\mid \boldsymbol{c}\in C\}$，显然，$C(x)=\Psi(C)$是 $F_q[x]_n$ 的一个线性子空间。

当 C 是循环码时，若 $\boldsymbol{c}=(c_0,c_1,\cdots,c_{n-1})\in C$，则 $\boldsymbol{c}'=(c_{n-1},c_0,c_1,\cdots,c_{n-2})\in C$，于是 $\boldsymbol{c}'$的码多项式为 $\boldsymbol{c}'(x)=c_{n-1}+c_0x+\cdots+c_{n-3}x^{n-2}+c_{n-2}x^{n-1}\in F_q[x]_n$。显然，$c'(x)\equiv xc(x)\bmod(x^n-1)$。若记以多项式 x^n-1 为模的剩余类环为 $F_q[x]/(x^n-1)$，则 $\boldsymbol{c}'$的码多项式 $c'(x)$恰好对应 $F_q[x]/(x^n-1)$中的元素 $xc(x)$。设 $g(x)\in F_q[x]$，由多项式的带余除法可知，$g(x)=h(x)(x^n-1)+r(x)$，$\deg r\leqslant n-1$，即$\overline{g(x)}=r(x)$，所以 $F_q[x]/(x^n-1)$中的每个元素$\overline{g(x)}$都有 $F_q[x]_n$ 中的唯一元素 $r(x)$作为其代表。因此，在模 x^n-1 的意义下，$F_q[x]/(x^n-1)=F_q[x]_n$。

定理 11.2.6 $C\subseteq F_q^n$ 是循环码当且仅当 $C(x)$是 $F_q[x]/(x^n-1)$中的理想。

证明： 假设 $C\subseteq F_q^n$ 是循环码。由 C 的线性性可知，C 对应的 $C(x)=\Psi(C)$是 $F_q[x]/(x^n-1)$的线性子空间。设 $c(x)\in C(x)$，由 C 的循环性可知，$c(x)$对应的码字 $\boldsymbol{c}$ 的任意循环移位都属于 C。特别地，循环右移 1 位、2 位、……、$n-1$ 位都属于 C，这些码字所对应的码多项式分别为 $xc(x)$、$x^2c(x)$、…、$x^{n-1}c(x)$，这些多项式当然都属于 $C(x)$。对任意的 $r(x)\in F_q[x]/(x^n-1)$，$r(x)$都可以表示为 1、x、x^2、…、x^{n-1}在 F_q 上的一个线性组合，即 $r(x)=r_0+r_1x+\cdots+r_{n-1}x^{n-1}$，$r_i\in F_q$，$i=0,1,\cdots,n-1$。因此，由 $C(x)$的线性性可知，$r(x)c(x)\in C(x)$。综上所述，$C(x)$是 $F_q[x]/(x^n-1)$中的理想。

反之，假设 $C(x)$是 $F_q[x]/(x^n-1)$中的理想，显然，$C(x)$对应的 $C=\Psi^{-1}(C(x))$是 F_q^n 的线性子空间即线性码。设 $\boldsymbol{c}\in C=\Psi^{-1}(C(x))$，则 $\boldsymbol{c}$ 的码多项式 $c(x)\in C(x)$。由于 $C(x)$是理想，所以 $xc(x)\in C(x)$。因此，$xc(x)$所对应的 $\boldsymbol{c}'\in C=\Psi^{-1}(C(x))$，而 $\boldsymbol{c}'$恰是 $\boldsymbol{c}$ 的循环右移，所以 $C\subseteq F_q^n$ 是循环码。

由近世代数知识可知，$F_q[x]/(x^n-1)$中的理想均是主理想，即理想中的每个元素都是由一个元素的倍式组成。既然 $C(x)$是 $F_q[x]/(x^n-1)$中的理想，所以 $C(x)$是一个主理想，必然能找到一个生成这个主理想的、次数最低的、最高次项系数等于 1(首一)的多项式 $g(x)$，使得 $C(x)=(g(x))=\{r(x)g(x)\in F_q[x]/(x^n-1)\mid r(x)\in F_q[x]/(x^n-1)\}$。把主理想 $C(x)$的生成元 $g(x)$称为循环码 C 的生成多项式，所有的码多项式都是 $g(x)$的倍式。由此可以直接证明以下结论。

定理 11.2.7 设 $C\subseteq F_q^n$ 是一个参数为$[n,k]$的循环码，则存在唯一的 $n-k$ 次首一多项式 $g(x)$使得每个码多项式 $c(x)\in C(x)$都是 $g(x)$的倍式，且每个低于 n 次的 $g(x)$的倍式一定是码多项式。

下面的定理给出了循环码的生成多项式应满足的条件。

定理 11.2.8 参数为$[n,k]$的循环码 $C\subseteq F_q^n$ 的生成多项式 $g(x)$一定是 x^n-1 的因式，即 $x^n-1=g(x)h(x)$。反之，若 $g(x)$是 x^n-1 的次数为 $n-k$ 的因式，则 $g(x)$一定能生成参数为$[n,k]$的循环码。

证明： 假设 $g(x)$不是 x^n-1 的因式，则由多项式的带余除法可知，$x^n-1=g(x)h(x)+r(x)$，$0<\deg r<\deg g$，由此可知，$r(x)=(x^n-1)-g(x)h(x)\in F_q[x]/(x^n-1)$，这说明在 $F_q[x]/(x^n-1)$中找到了比 $g(x)$的次数更低的多项式 $r(x)$，导致

矛盾。这就证明了 $g(x)$ 一定是 x^n-1 的因式。剩余的结论可由前面的讨论容易证明。

定理 11.2.8 说明，要寻找一个参数为 $[n,k]$ 的循环码，就是要寻找一个能除尽 x^n-1 的 $n-k$ 次首一多项式 $g(x)$，由 $g(x)$ 生成的主理想就是一个参数为 $[n,k]$ 的循环码。这说明，只要知道了 x^n-1 的因式分解，用它的各个因式的乘积便可构造出许多不同的循环码。

例 11.2.11　构造一个参数为 $[7,3]$ 的二元循环码。由于在 F_2 上 $x^7-1=(x+1)(x^3+x+1)(x^3+x^2+1)$，取 $g(x)=(x+1)(x^3+x+1)$，由 $g(x)$ 生成的循环码便是一个参数为 $[7,3]$ 的二元循环码。

接下来讨论循环码的矩阵表示。

设参数为 $[n,k]$ 的循环码 $C\subseteq F_q^n$ 的生成多项式是 $g(x)$，则由定理 11.2.8 可知，$x^n-1=g(x)h(x)$，$\deg g=n-k$，$\deg h=k$。由定理 11.2.7 易知，$g(x)$、$xg(x)$、…、$x^{k-1}g(x)$ 是 $C(x)$ 在 F_q 上的一组基，其线性组合可以把所有的 q^k 个码多项式产生出来。因此，这组基所对应的 k 个 n 重向量作为行所构成的 $k\times n$ 阶矩阵 $\boldsymbol{G}$ 是循环码 C 的生成矩阵。所以，可以用以下方法得到生成矩阵 $\boldsymbol{G}$。

设 $g(x)=g_0+g_1x+\cdots+g_{n-k-1}x^{n-k-1}+g_{n-k}x^{n-k}$，$g_{n-k}=1$，则

$$\begin{aligned} xg(x) &= g_0x+g_1x^2+\cdots+g_{n-k-1}x^{n-k}+g_{n-k}x^{n-k+1} \\ &\vdots \\ x^{k-1}g(x) &= g_0x^{k-1}+g_1x^k+\cdots+g_{n-k-1}x^{n-2}+g_{n-k}x^{n-1} \end{aligned}$$

因此，码 C 的生成矩阵多项式为

$$\boldsymbol{G}(x)=\begin{pmatrix} g(x) \\ \vdots \\ x^{k-2}g(x) \\ x^{k-1}g(x) \end{pmatrix}$$

相应地，码 C 的生成矩阵为

$$\boldsymbol{G}=\begin{pmatrix} g_0 & g_1 & \cdots & g_{n-k-1} & g_{n-k} & 0 & 0 & \cdots & 0 \\ 0 & g_0 & g_1 & \cdots & g_{n-k-1} & g_{n-k} & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \cdots & 0 & g_0 & g_1 & \cdots & \cdots & g_{n-k-1} & g_{n-k} \end{pmatrix}$$

设 $h(x)=h_0+h_1x+\cdots+h_{k-1}x^{k-1}+h_kx^k$，由 $x^n-1=g(x)h(x)=(g_0+g_1x+\cdots+g_{n-k-1}x^{n-k-1}+g_{n-k}x^{n-k})(h_0+h_1x+\cdots+h_{k-1}x^{k-1}+h_kx^k)$ 及 $x,x^2,\cdots,x^{n-1}$ 系数均为 0 的事实可知

$$\begin{aligned} g_{n-k}h_k &= 1 \\ g_0h_0 &= -1 \\ g_0h_1+g_1h_0 &= 0 \\ g_0h_2+g_1h_1+g_2h_0 &= 0 \\ &\vdots \\ g_{n-1}h_0+g_{n-2}h_1+\cdots+g_{n-k}h_{k-1} &= 0 \end{aligned}$$

这里约定 $g_i=0(i=n-k+1,n-k+2,\cdots,n-1)$，$h_j=0(j=k+1,k+2,\cdots,n-1)$。

由此可知，码 C 的校验矩阵为

$$\boldsymbol{H}=\begin{pmatrix} h_k & h_{k-1} & \cdots & h_1 & h_0 & 0 & 0 & \cdots & 0 \\ 0 & h_k & h_{k-1} & \cdots & h_1 & h_0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \cdots & 0 & h_k & h_{k-1} & \cdots & \cdots & h_1 & h_0 \end{pmatrix}$$

$\boldsymbol{H}$ 完全由 $h(x)$ 的系数决定，因此，也称 $h(x)$ 是循环码 C 的校验多项式。

可直接验证 $\boldsymbol{G}\cdot\boldsymbol{H}^{\mathrm{T}}=\boldsymbol{0}$，式中的 $\boldsymbol{0}$ 是一个 $k\times(n-k)$ 阶零矩阵。由 $g_{n-k}h_k=1$，$g_0h_0=-1$ 可知，$\boldsymbol{G}$ 和 $\boldsymbol{H}$ 的秩分别为 k 和 $n-k$，并且 $h_k=1$（因 $g_{n-k}=1$），g_0 和 h_0 均不为零。

由 11.2.3 小节中的讨论可知，循环码 C 的对偶码 $C^{\perp}$ 以 $\boldsymbol{H}$ 和 $\boldsymbol{G}$ 分别为生成矩阵和校验矩阵。由 $\boldsymbol{G}$ 和 $\boldsymbol{H}$ 的形式可以看出，以 $h_0^{-1}x^kh(x^{-1})$ 为生成多项式的循环码和以 $g(x)$ 为生成多项式的循环码互为对偶码。

例 11.2.12 例 11.2.11 构造的参数为[7,3]的二元循环码中，$g(x)=(x+1)(x^3+x+1)$，$h(x)=x^3+x^2+1$，则

$$\boldsymbol{G}=\begin{pmatrix} 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{pmatrix},\quad \boldsymbol{H}=\begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \end{pmatrix}$$

最后讨论循环码的根表示。

设 $C\subseteq F_q^n$ 是一个参数为$[n,k]$的循环码，其生成多项式是 $g(x)$，则将 $g(x)$ 在 F_q 的扩域中的根叫做循环码 C 的根。

为了便于讨论以下假设 n 与 q 互素。此时 x^n-1 在 F_q 的扩域中无重根，而 $g(x)\mid(x^n-1)$，所以 $g(x)$ 在 F_q 的扩域中也无重根。记 $g(x)$ 在 F_q 的扩域中的所有根组成的集合为 T，$|T|=\deg g=n-k$，则对任意的 $c(x)\in F_q[x]$，有

$c(x)$ 对应的 $\boldsymbol{c}\in F_q^n$ 为 C 的码字当且仅当 $g(x)\mid c(x)$ 当且仅当对每个 $\gamma\in T$，$c(\gamma)=0$ 所以循环码 C 可以通过以下方式来表示：

$$\{c(x)\in F_q[x]\mid \deg c(x)\leqslant n-1,\forall\gamma\in T,c(\gamma)=0\}$$

用这种根的表示方式可以给出循环码 C 的一个校验矩阵。因为 $\boldsymbol{c}=(c_0,c_1,\cdots,c_{n-1})\in C$ 当且仅当 $c(x)=c_0+c_1x+\cdots+c_{n-1}x^{n-1}\in C(x)$ 当且仅当对每个 $\gamma_i\in T$，$i=1,2,\cdots,|T|=n-k$，$c(\gamma)=c_0+c_1\gamma_i+\cdots+c_{n-1}\gamma_i^{n-1}=0$ 当且仅当 $\boldsymbol{c}\boldsymbol{H}^{\mathrm{T}}=\boldsymbol{0}$。其中

$$\boldsymbol{H}=\begin{pmatrix} 1 & \gamma_1 & \cdots & \gamma_1^{n-1} \\ 1 & \gamma_2 & \cdots & \gamma_2^{n-1} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & \gamma_{n-k} & \cdots & \gamma_{n-k}^{n-1} \end{pmatrix}$$

例 11.2.13 考虑参数为[7,4]的二元循环码，取 $g(x)=1+x+x^3$，它以 F_{2^3} 的

本原元 α、α^2 及 α^4 为根，则它的校验矩阵为

$$\boldsymbol{H}=\begin{pmatrix}1 & \alpha & \alpha^2 & \alpha^3 & \alpha^4 & \alpha^5 & \alpha^6\\ 1 & \alpha^2 & \alpha^4 & \alpha^6 & \alpha & \alpha^3 & \alpha^5\\ 1 & \alpha^4 & \alpha & \alpha^5 & \alpha^2 & \alpha^6 & \alpha^3\end{pmatrix}$$

根据 F_{2^3} 的本原元表示与 F_2^3 之间的对应关系，可将上述矩阵化为二元矩阵，再删去线性相关的行向量，最终可得到[7,4]二元循环码的一个二元校验矩阵。

有了上述 3 种表示方式，可以用其中任何一种方式表示循环码。也可以把 $\boldsymbol{c}$ 和 $c(x)$看成一样的。常用的循环码主要有两种，即 BCH 码和 Reed-Solomon 码，这两种码将在下节介绍。

11.3　一些好码

寻找好码是纠错码中的一个核心研究课题，目前已经构造出了很多好码，但限于篇幅和写作本书的目的，本节只介绍在信息安全中最常用的几类典型好码，包括 BCH 码、广义 Reed-Solomon 码（含 Reed-Solomon 码和扩展 Reed-Solomon 码）、Goppa 码和二元 Reed-Muller 码。

11.3.1　BCH 码

BCH 码是一类分别由 Hocquenghen 于 1959 年，以及 Bose 和 Chaudhuri 于 1960 年独立提出的循环码，有着丰富的代数结构、深入的研究结果和广泛的应用。

定义 11.3.1　设 n 与 q 互素（这个条件主要是为了保证在 F_q 的扩域 F_{q^m} 中存在阶为 n 的元素 α，此时 $\beta^{(q^m-1)/n}$的阶就是 n，β 是 F_{q^m} 的本原元），l 和 δ 是正整数，$2\leqslant\delta\leqslant n-1$，则称以 $\delta-1$ 个连续的方幂 $\alpha^l,\alpha^{l+1},\cdots,\alpha^{l+\delta-2}$ 为根的码 $C=\{c(x)\in F_q[x]/(x^n-1)\mid c(\alpha^{l+i})=0,i=0,1,\cdots,\delta-2\}$是设计距离为 δ 的 BCH 码。

当 $n=q^m-1$ 时，称为本原 BCH 码；否则，称为非本原 BCH 码。

特别地，当取 $n=q-1$ 时，此时 n 阶元素 α 就是 F_q 的本原元，这类循环码称为 Reed-Solomon 码，由于这类码是广义 Reed-Solomon 码的一种特殊情况，所以将在下节中讨论。但值得注意的是，通常广义 Reed-Solomon 码是一类线性码，而 Reed-Solomon 码是一类循环码。

注意到 $c(x)$与 $xc(x)$一定有相同的非零根，并由定义直接可得出下面的定理。

定理 11.3.1　BCH 码是一个参数为$[n,k]$的循环码，其生成多项式 $g(x)=\mathrm{Lcm}\{g_0(x),g_1(x),\cdots,g_{\delta-2}(x)\}$，其中 $g_i(x)(i=0,1,\cdots,\delta-2)$表示 α^{l+i}的极小多项式，$\mathrm{Lcm}\{g_0(x),g_1(x),\cdots,g_{\delta-2}(x)\}$表示 $g_0(x),g_1(x),\cdots,g_{\delta-2}(x)$的最小公倍式，$k=n-\deg g$。

下面的定理给出了 BCH 码的极小距离 d 与其设计距离 δ 之间的关系。

定理 11.3.2　BCH 码的极小距离 d 不小于设计距离 δ，即 $d\geqslant\delta$。

证明：设 $c(x)=c_0+c_1x+\cdots+c_{n-1}x^{n-1}\in C$，则由定义可知，$c(x)\in C$ 当且仅当

$c(\alpha^{l+i})=c_0+c_1\alpha^{l+i}+\cdots+c_{n-1}\alpha^{(l+i)(n-1)}=0, i=0,1,\cdots,\delta-2$ 当且仅当 $\boldsymbol{cH}^{\mathrm{T}}=\mathbf{0}$。其中

$$\boldsymbol{H}=\begin{pmatrix} 1 & \alpha^l & \cdots & (\alpha^l)^{n-1} \\ 1 & \alpha^{l+1} & \cdots & (\alpha^{l+1})^{n-1} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & \alpha^{l+\delta-2} & \cdots & (\alpha^{l+\delta-2})^{n-1} \end{pmatrix}$$

由推论 11.2.1 可知，欲证 $d\geqslant\delta$，只需证明 $\boldsymbol{H}$ 的任意 $\delta-1$ 列都在 F_q 上线性独立，因此，只需证明 $\boldsymbol{H}$ 的任意 $\delta-1$ 列组成的 $(\delta-1)\times(\delta-1)$ 阶矩阵的行列式都不等于零。取 $\boldsymbol{H}$ 的任意 $\delta-1$ 列，给出下列 $(\delta-1)\times(\delta-1)$ 阶矩阵，即

$$\boldsymbol{M}=\begin{pmatrix} \alpha^{i_1 l} & \alpha^{i_2 l} & \cdots & \alpha^{i_{\delta-1} l} \\ \alpha^{i_1(l+1)} & \alpha^{i_2(l+1)} & \cdots & \alpha^{i_{\delta-1}(l+1)} \\ \vdots & \vdots & \ddots & \vdots \\ \alpha^{i_1(l+\delta-2)} & \alpha^{i_2(l+\delta-2)} & \cdots & \alpha^{i_{\delta-1}(l+\delta-2)} \end{pmatrix}, \quad 0\leqslant i_1<i_2<\cdots<i_{\delta-1}\leqslant n-1$$

则有

$$|\boldsymbol{M}|=\begin{vmatrix} \alpha^{i_1 l} & \alpha^{i_2 l} & \cdots & \alpha^{i_{\delta-1} l} \\ \alpha^{i_1(l+1)} & \alpha^{i_2(l+1)} & \cdots & \alpha^{i_{\delta-1}(l+1)} \\ \vdots & \vdots & \ddots & \vdots \\ \alpha^{i_1(l+\delta-2)} & \alpha^{i_2(l+\delta-2)} & \cdots & \alpha^{i_{\delta-1}(l+\delta-2)} \end{vmatrix}$$

$$=\alpha^{(i_1+i_2+\cdots+i_{\delta-1})l}\begin{vmatrix} 1 & 1 & \cdots & 1 \\ \alpha^{i_1} & \alpha^{i_2} & \cdots & \alpha^{i_{\delta-1}} \\ \vdots & \vdots & \ddots & \vdots \\ \alpha^{i_1(\delta-2)} & \alpha^{i_2(\delta-2)} & \cdots & \alpha^{i_{\delta-1}(\delta-2)} \end{vmatrix}$$

因为 $\alpha^{i_1},\alpha^{i_2},\cdots,\alpha^{i_{\delta-1}}$ 互不相同且均不为零，上述右边的行列式是 Vandermonde 行列式，所以 $|\boldsymbol{M}|\neq 0$，这就证明了 $d\geqslant\delta$。

例 11.3.1 设 α 是 F_{2^4} 的本原元，取 $n=15$，若 α 是 BCH 码 C 的根，则其共轭 α^2、α^4、α^8 也都是码 C 的根，其中有两个连续的方幂 α、α^2，因此，码 C 的设计距离 $\delta=3$。α 的极小多项式是 x^4+x+1，所以码 C 的生成多项式 $g(x)=x^4+x+1$，$k=n-\deg g=15-4=11$。因此得到一个参数为[15,11,3]的 BCH 码。

11.3.2 广义 Reed-Solomon 码

Reed-Solomon 码最早是由 Reed 和 Solomon 于 1960 年构造出来的一类 MDS 码，而广义 Reed-Solomon 码是 Reed-Solomon 码的一种推广，很多常用的线性码(如 Reed-Solomon 码、扩展 Reed-Solomon 和 Goppa 码)都可以由一个广义 Reed-Solomon 码来表示。它不仅在各种通信系统中得到普遍使用，而且在存储系统如 CD 或 DVD 中也大量使用，并且在信息安全领域中得到广泛使用，如用于密码算法设计、认证系统的构造和秘密共享协议的设计等。

定义 11.3.2 设 F 是一个域，n 是正整数，$0\leqslant k\leqslant n$ 且 k 是整数。假设 $v_1,v_2,\cdots,v_n$ 是 F 中的非零元，并且 $\alpha_1,\alpha_2,\cdots,\alpha_n$ 是 F 中两两互不相同的元素。令 $\boldsymbol{\alpha}=(\alpha_1,\alpha_2,\cdots,\alpha_n)\in F^n$，$\boldsymbol{v}=(v_1,v_2,\cdots,v_n)\in F^n$，广义 Reed-Solomon 码定义为

$$\mathrm{GRS}_{n,k}(\boldsymbol{\alpha},\boldsymbol{v})=\{(v_1f(\alpha_1),v_2f(\alpha_2),\cdots,v_nf(\alpha_n))\mid f(x)\in F[x]_k\}\subseteq F^n$$

有时也把$(v_1f(\alpha_1),v_2f(\alpha_2),\cdots,v_nf(\alpha_n))$简记为 $\boldsymbol{f}$。

例 11.3.2 设 $F=F_3=\ _3$，$n=3$，$k=2$，$\boldsymbol{v}=(111)$，$\boldsymbol{\alpha}=(012)$。则

$$\mathrm{GRS}_{3,2}(\boldsymbol{\alpha},\boldsymbol{v})=\{(f(0),f(1),f(2))\mid f(x)\in F_3[x]_2\}。$$

其中

$$F_3[x]_2=\{0,1,2,x,x+1,x+2,2x,2x+1,2x+2\}$$

所以

$$\mathrm{GRS}_{3,2}(\boldsymbol{\alpha},\boldsymbol{v})=\{(000),(111),(222),(012),(120),(201)(021),(102)\}\subseteq F_3^3$$

显然，域 F 的大小对 $\mathrm{GRS}_{n,k}(\boldsymbol{\alpha},\boldsymbol{v})$有必然的影响。特别地，由于向量$\boldsymbol{\alpha}$ 中的分量不同，n 不会比域 F 中的元素数目大。例如，若 $F=F_2$，则 n 最大只能取到 2。当然，长为 2 的码没有特别的意义，所以人们很少在广义 Reed-Solomon 码中令 $F=F_2$。

关于码 $\mathrm{GRS}_{n,k}(\boldsymbol{\alpha},\boldsymbol{v})$，容易证明以下定理。

定理 11.3.3 $\mathrm{GRS}_{n,k}(\boldsymbol{\alpha},\boldsymbol{v})$是线性码。

证明 $\mathrm{GRS}_{n,k}(\boldsymbol{\alpha},\boldsymbol{v})$的一些性质，需要引入下面的引理。

引理 11.3.1 设 F 是一个域，k 是一个正整数，$h(x)$是 $F[x]_k$ 中的一个多项式，则 $h(x)$在 F 中至多有 $k-1$ 个根。

由定理 2.3.2 余式定理可直接证明引理 11.3.1。

定理 11.3.4 设 F_q 是一个有限域，n 是正整数，$0\leqslant k\leqslant n$ 且 k 是整数，则 $\mathrm{GRS}_{n,k}(\boldsymbol{\alpha},\boldsymbol{v})$是一个参数为$[n,k]$的线性码，并且若 $k\neq0$，则 $\mathrm{GRS}_{n,k}(\boldsymbol{\alpha},\boldsymbol{v})$是 MDS 码。

证明：令 $C=\mathrm{GRS}_{n,k}(\boldsymbol{\alpha},\boldsymbol{v})$，由定理 11.3.3 可知，$C$ 是线性码。为了证明 C 的维数是 k，首先证明$|C|=|F_q|^k=q^k$，其中$|C|$表示 C 中的元素个数，$|F_q|$表示 F_q 中的元素个数。易知，$F_q[x]_k$ 中的元素个数是$|F_q|^k=q^k$，所以有$|C|\leqslant q^k$。假设$|C|<q^k$，则存在多项式 $f(x),g(x)\in F_q[x]_k$ 使得 $f(x)\neq g(x)$，但是 $\boldsymbol{f}=\boldsymbol{g}$。令 $h(x)=f(x)-g(x)\in F_q[x]_k$，且 $h(x)$不为零多项式，有

$$\begin{aligned}\boldsymbol{h}&=(v_1h(\alpha_1),\cdots,v_nh(\alpha_n))\\&=(v_1f(\alpha_1),\cdots,v_nf(\alpha_n))-(v_1g(\alpha_1),\cdots,v_ng(\alpha_n))\\&=\boldsymbol{f}-\boldsymbol{g}=\boldsymbol{0}\end{aligned}$$

所以，对 $i=1,2,\cdots,n$，都有 $v_ih(\alpha_i)=0$，但是对所有的 i，有 $v_i\neq0$，因此，对 $i=1,2,\cdots,n$，必须有 $h(\alpha_i)=0$。因为这些 α_i 是两两互不相同的，所以 $h(x)$在 F 中有 n 个根。但是由引理 11.3.1 可知，$h(x)$在 F 中至多有 $k-1<n$ 个根，矛盾。于是有$|C|=|F_q|^k=q^k$。

现在来证明 C 的维数一定是 k。显然 C 的维数至少是 k（否则 C 的元素个数小于$|F_q|^k=q^k$）。假设 C 的维数是 l，其中 $l>k$，则存在$\boldsymbol{v}_1,\cdots,\boldsymbol{v}_l\in C$ 满足$\boldsymbol{v}_1,\cdots,\boldsymbol{v}_l$ 线性独立，所以若矩阵 $\boldsymbol{G}$ 的第 i 行是$\boldsymbol{v}_i$，则 $C=\mathrm{RS}(\boldsymbol{G})$。也即是说，$\boldsymbol{G}$ 是 C 的一个生成矩阵，即对所有的 $c_i\in F_q$，$i=1,2,\cdots,l$，有 $c_1\boldsymbol{v}_1+c_2\boldsymbol{v}_2+\cdots+c_l\boldsymbol{v}_l\in C$。若所有具有这种形式的码字互不相同，则 C 将会有$|F_q|^l>|F_q|^k$ 个元素，与$|C|=|F_q|^k$ 矛盾。于是对 $i=1,2,\cdots,l$ 存在 $c_i,d_i\in F_q$，使得对某个 j，$c_j\neq d_j$，而且

$$c_1\boldsymbol{v}_1+c_2\boldsymbol{v}_2+\cdots+c_l\boldsymbol{v}_l=d_1\boldsymbol{v}_1+d_2\boldsymbol{v}_2+\cdots+d_l\boldsymbol{v}_l$$

所以有

$$(c_1-d_1)\boldsymbol{v}_1+(c_2-d_2)\boldsymbol{v}_2+\cdots+(c_l-d_l)\boldsymbol{v}_l=\boldsymbol{0}$$

但是 $c_j-d_j\neq 0$，所以 $\boldsymbol{v}_1,\cdots,\boldsymbol{v}_l$ 不是线性独立的，推出矛盾。于是，C 的维数是 k。

最后证明若 $k\neq 0$，则 C 是 MDS 码。由 Singleton 界，有 $d_{\min}(C)\leqslant n-k+1$。只需证明 $d_{\min}(C)\geqslant n-k+1$ 即可。对所有的 $\boldsymbol{h}\in C$ 且 $\boldsymbol{h}\neq\boldsymbol{0}$（因为 $k\neq 0$，所以这样的码字 $\boldsymbol{h}$ 一定存在），由引理 11.3.1 可知，$h(x)\in F_q[x]_k$ 至多有 $k-1$ 个根，于是 $\boldsymbol{h}$ 的至多 $k-1$ 个分量为 0，所以 $W_H(\boldsymbol{h})\geqslant n-(k-1)=n-k+1$，于是有 $d_{\min}(C)\geqslant n-k+1$，因此，$C$ 是 MDS 码。

定理 11.3.4 表明，广义 Reed-Solomon 码就纠错而言是一个好码。此外，也证明了 $|C|=|F_q|^k$。于是 $F_q[x]_k$ 中的不同元素给出了 C 中的不同码字，也即是说，若 $f(x),g(x)\in F_q[x]_k$ 使得 $f(x)\neq g(x)$，则 $\boldsymbol{f}\neq\boldsymbol{g}$。

由于 $\mathrm{GRS}_{n,k}(\boldsymbol{\alpha},\boldsymbol{v})$ 是一个参数为 $[n,k]$ 的线性码，它必然有一个生成矩阵 $\boldsymbol{G}$，定理 11.3.5 给出了其中的一个。

定理 11.3.5 矩阵

$$\boldsymbol{G}=\begin{pmatrix} v_1 & v_2 & \cdots & v_n \\ v_1\alpha_1 & v_2\alpha_2 & \cdots & v_n\alpha_n \\ \vdots & \vdots & & \vdots \\ v_1\alpha_1^i & v_2\alpha_2^i & \cdots & v_n\alpha_n^i \\ \vdots & \vdots & & \vdots \\ v_1\alpha_1^{k-1} & v_2\alpha_2^{k-1} & \cdots & v_n\alpha_n^{k-1} \end{pmatrix}$$

是 $\mathrm{GRS}_{n,k}(\boldsymbol{\alpha},\boldsymbol{v})$ 的一个生成矩阵。

证明：需要证明两点：① $\boldsymbol{G}$ 的行线性独立；② $\mathrm{GRS}_{n,k}(\boldsymbol{\alpha},\boldsymbol{v})=\mathrm{RS}(\boldsymbol{G})$。

先证明 $\boldsymbol{G}$ 的行线性独立。假设

$$c_0(v_1,\cdots,v_n)+c_1(v_1\alpha_1,\cdots,v_n\alpha_n)+\cdots+c_{k-1}(v_1\alpha_1^{k-1},\cdots,v_n\alpha_n^{k-1})=(0,0,\cdots,0)$$

则下面的等式成立

$$v_1(c_0+c_1\alpha_1+c_2\alpha_1^2+\cdots+c_{k-1}\alpha_1^{k-1})=0$$
$$v_2(c_0+c_1\alpha_2+c_2\alpha_2^2+\cdots+c_{k-1}\alpha_2^{k-1})=0$$
$$\vdots$$
$$v_n(c_0+c_1\alpha_n+c_2\alpha_n^2+\cdots+c_{k-1}\alpha_n^{k-1})=0$$

但是对所有的 i，$v_i\neq 0$，所以对所有的 $i=1,2,\cdots,n$，α_i 是多项式 $f(x)=c_0+c_1x+c_2x^2+\cdots+c_{k-1}x^{k-1}\in F_q[x]_k$ 的根。但是 $k\leqslant n$，若 $f(x)$ 不是零多项式，则由引理 11.3.1，它至多有 $k-1$ 个不同的根，于是 $f(x)$ 一定是零多项式，所以 $c_0=0,c_1=0,\cdots,c_k=0$，于是 $\boldsymbol{G}$ 的行是线性独立的。

现在证明 $\mathrm{GRS}_{n,k}(\boldsymbol{\alpha},\boldsymbol{v})=\mathrm{RS}(\boldsymbol{G})$。令 $\boldsymbol{f}\in\mathrm{GRS}_{n,k}(\boldsymbol{\alpha},\boldsymbol{v})$，则存在 $f(x)\in F_q[x]_k$ 使得 $\boldsymbol{f}=(v_1f(\alpha_1),v_2f(\alpha_2),\cdots,v_nf(\alpha_n))$。令 $f(x)=c_0+c_1x+c_2x^2+\cdots+c_{k-1}x^{k-1}$，则

$$\boldsymbol{f}=(v_1(c_0+c_1\alpha_1+c_2\alpha_1^2+\cdots+c_{k-1}\alpha_1^{k-1}),\\ v_2(c_0+c_1\alpha_2+c_2\alpha_2^2+\cdots+c_{k-1}\alpha_2^{k-1}),\\ \cdots,\\ v_n(c_0+c_1\alpha_n+c_2\alpha_n^2+\cdots+c_{k-1}\alpha_n^{k-1}))$$

进行重组后,可以得到

$$\boldsymbol{f}=(v_1c_0,v_2c_0,\cdots,v_nc_0)+(v_1c_1\alpha_1,v_2c_1\alpha_2,\cdots,v_2c_1\alpha_2)\\ +\cdots+(v_1c_{k-1}\alpha_1^{k-1},v_2c_{k-1}\alpha_2^{k-1},\cdots,v_nc_{k-1}\alpha_n^{k-1})$$

所以

$$\boldsymbol{f}=c_0(v_1,\cdots,v_n)+c_1(v_1\alpha_1,\cdots,v_n\alpha_n)+\cdots+c_{k-1}(v_1\alpha_1^{k-1},\cdots,v_n\alpha_n^{k-1})。$$

所以 $\boldsymbol{f}\in \mathrm{RS}(\boldsymbol{G})$,因此,$\mathrm{GRS}_{n,k}(\boldsymbol{\alpha},\boldsymbol{v})\subseteq \mathrm{RS}(\boldsymbol{G})$,又 $\mathrm{GRS}_{n,k}(\boldsymbol{\alpha},\boldsymbol{v})$ 和 $\mathrm{RS}(\boldsymbol{G})$ 的维数是一样的,所以 $\mathrm{GRS}_{n,k}(\boldsymbol{\alpha},\boldsymbol{v})=\mathrm{RS}(\boldsymbol{G})$。

定理 11.3.2 表明,$\mathrm{GRS}_{n,k}(\boldsymbol{\alpha},\boldsymbol{v})$ 是一个参数为 $[n,k]$ 的线性码,所以,由定理 11.2.3可知,$\mathrm{GRS}_{n,k}(\boldsymbol{\alpha},\boldsymbol{v})$ 的对偶码是一个参数为 $[n,n-k]$ 的线性码。下面定理 11.3.6给出了 $\mathrm{GRS}_{n,k}(\boldsymbol{\alpha},\boldsymbol{v})$ 的对偶码的结构。

定理 11.3.6 $\mathrm{GRS}_{n,k}(\boldsymbol{\alpha},\boldsymbol{v})^{\perp}=\mathrm{GRS}_{n,n-k}(\boldsymbol{\alpha},\boldsymbol{u})$,其中 $\boldsymbol{u}=(u_1,u_2,\cdots,u_n)$,$u_j=v_j^{-1}\left(\prod_{i=1,i\neq j}^{n}(\alpha_j-\alpha_i)\right)^{-1}$,$j=1,2,\cdots,n$。

证明:定义

$$L(x)=\prod_{i=1}^{n}(x-\alpha_i)$$

且

$$L_j(x)=\prod_{i=1,i\neq j}^{n}(x-\alpha_i)$$

首先证明,若 $\boldsymbol{f}\in \mathrm{GRS}_{n,k}(\boldsymbol{\alpha},\boldsymbol{v})$,$\boldsymbol{g}\in \mathrm{GRS}_{n,n-k}(\boldsymbol{\alpha},\boldsymbol{u})$,则 $\boldsymbol{f}$ 和 $\boldsymbol{g}$ 的点积为 0。令 $f(x)\in F_q[x]_k$ 为 $\boldsymbol{f}$ 对应的多项式,$g(x)\in F_q[x]_{n-k}$ 为 $\boldsymbol{g}$ 对应的多项式。乘积 $f(x)g(x)$ 的次数至多为 $n-2$。对每个 $i=1,2,\cdots,n$,有

$$\sum_{j=1}^{n}\frac{L_j(\alpha_i)}{L_j(\alpha_j)}f(\alpha_j)g(\alpha_j)=\frac{L_i(\alpha_i)}{L_i(\alpha_i)}f(\alpha_i)g(\alpha_i)=f(\alpha_i)g(\alpha_i)$$

容易证明

$$f(x)g(x)=\sum_{j=1}^{n}\frac{L_j(x)}{L_j(\alpha_j)}f(\alpha_j)g(\alpha_j)$$

$f(x)g(x)$ 的次数不大于 $n-2$,因此等式左边的 x^{n-1} 项的系数为 0,等式右边的 x^{n-1} 项的系数也为 0。所以,

$$0=\sum_{j=1}^{n}\frac{1}{L_j(\alpha_j)}f(\alpha_j)g(\alpha_j)=\sum_{j=1}^{n}(v_jf(\alpha_j))(u_jg(\alpha_j))$$

这恰好是 $\boldsymbol{f}$ 和 $\boldsymbol{g}$ 的点积。所以,$\boldsymbol{f}$ 和 $\boldsymbol{g}$ 的点积为 0,于是有 $\mathrm{GRS}_{n,n-k}(\boldsymbol{\alpha},\boldsymbol{u})\subseteq \mathrm{GRS}_{n,k}(\boldsymbol{\alpha},\boldsymbol{v})^{\perp}$。但是由于它们都是维数为 $n-k$ 的线性码,所以等式成立。

因为一个线性码的校验矩阵是它的对偶码的生成矩阵,所以可以很容易地写出 $\mathrm{GRS}_{n,k}(\boldsymbol{\alpha},\boldsymbol{v})$ 的一个校验矩阵 $\boldsymbol{H}$。

$$\boldsymbol{H}=\begin{pmatrix} u_1 & u_2 & \cdots & u_n \\ u_1\alpha_1 & u_2\alpha_2 & \cdots & u_n\alpha_n \\ \vdots & \vdots & & \vdots \\ u_1\alpha_1^i & u_2\alpha_2^i & \cdots & u_n\alpha_n^i \\ \vdots & \vdots & & \vdots \\ u_1\alpha_1^{n-k-1} & u_2\alpha_2^{n-k-1} & \cdots & u_n\alpha_n^{n-k-1} \end{pmatrix}$$

特别地，在定义 11.3.2 中，设 F_q 是一个有限域，α 是 F_q 的本原元(也称生成元)，此时，F_q 中的元素可表示为 $\{0,1,\alpha,\cdots,\alpha^{q-2}\}$。当取 $n=|F_q|-1=q-1$，$v_1=v_2=\cdots=v_n=1$，$0\leqslant k\leqslant n$ 且 k 是整数，$\alpha_1=1,\alpha_2=\alpha,\cdots,\alpha_n=\alpha^{n-1}$，即可得到 Reed-Solomon 码的一个等价定义。当取 $n=|F_q|=q$，$v_1=v_2=\cdots=v_n=1$，$0\leqslant k\leqslant n$ 且 k 是整数，$\alpha_1=0,\alpha_2=1,\cdots,\alpha_n=\alpha^{n-2}$，即可得到扩展 Reed-Solomon 码的一个等价定义。当然，关于广义 Reed-Solomon 码的有关结果对 Reed-Solomon 码和扩展 Reed-Solomon 码也都是成立的。

11.3.3 Goppa 码

Goppa 码是俄国学者 Goppa 于 20 世纪 70 年代初提出的一类有理分式码，其主要优点是它的某些子类能达到 Shannon 信道编码定理所给出的性能，并且有快速译码算法。这类码可用于信息安全领域的研究中，如可用它构造各种密码算法和认证协议等。

定义 11.3.3 设 F_q 是一个有限域，$L=\{\alpha_1,\alpha_2,\cdots,\alpha_n\}$，$\alpha_i\in F_{q^m}(i=1,2,\cdots,n)$，$\alpha_i\neq\alpha_j(i\neq j)$。设 $g(x)\in F_{q^m}[x]$，$g(x)$ 的根不在 L 中，即 $g(\alpha_i)\neq 0(i=1,2,\cdots,n)$，Goppa 码定义为

$$C=\left\{(c_1,c_2,\cdots,c_n)\in F_q^n \,\middle|\, \sum_{i=1}^{n}\frac{c_i}{x-\alpha_i}\equiv 0 \bmod g(x)\right\}$$

简记为 $\Gamma(L,g)$，$g(x)$ 称为 Goppa 多项式。特别地，当 $g(x)$ 为不可约多项式时，称为不可约 Goppa 码。

用定义直接可验证，Goppa 码是一个线性码。另外，值得一提的是定义中使用了两个域，要注意元素的取值域。

下面的定理揭示了 Goppa 码和广义 Reed-Solomon 码之间的关系。

定理 11.3.7 Goppa 码 $\Gamma(L,g)$ 是一个广义 Reed-Solomon 码的子域子码。

证明：首先需要找出 Goppa 码 $\Gamma(L,g)$ 的校验矩阵。设 $g(x)=\sum_{i=1}^{t}g_ix^i$，于是

$$\phi(x)=\frac{g(x)-g(z)}{x-z}=\sum_{0\leqslant l+j\leqslant t-1}g_{l+j+1}z^jx^l$$

是关于变量 x 的一个次数小于 t 的多项式(对任何 z)。由于

$$\frac{1}{x-\alpha_i}\equiv\frac{-1}{g(\alpha_i)}\left[\frac{g(x)-g(\alpha_i)}{x-\alpha_i}\right] \bmod g(x)$$

从而依据 $(x-z)\phi(x)=g(x)-g(z)\equiv -g(z) \bmod g(x)$，再令 $\xi_j=g(\alpha_j)^{-1}(j=1,2,$

$\cdots,n$)，关系式

$$\sum_{i=1}^{n}\frac{c_i}{x-\alpha_i}\equiv 0 \bmod g(x)$$

可以改写成

$$\sum_{i=1}^{n}c_i\xi_i\sum_{0\leqslant l+j\leqslant t-1}g_{l+j+1}(\alpha_i)^j x^l\equiv 0 \bmod g(x)$$

对 $0\leqslant l\leqslant t-1$，上式中 x^l 的系数均为 0。可以发现，$c=(c_1,c_2,\cdots,c_n)$必与下列矩阵的每一行向量之内积为 0：

$$\begin{bmatrix} \xi_1 g_t & \cdots & \xi_n g_t \\ \xi_1(g_{t-1}+g_t\alpha_1) & \cdots & \xi_n(g_{t-1}+g_t\alpha_n) \\ \vdots & \ddots & \vdots \\ \xi_1(g_1+g_2\alpha_1+\cdots+g_t\alpha_1^{t-1}) & \cdots & \xi_n(g_1+g_2\alpha_n+\cdots+g_t\alpha_n^{t-1}) \end{bmatrix}$$

其中第 1 行对应于上述和式中 x^{t-1} 的系数，第 2 行对应于 x^{t-2} 的系数，……，第 t 行对应于 x^0 的系数。

由线性代数知识可知，对上述矩阵做初等行变换并不影响码 C 与该矩阵行的正交性。由此可以得到 Goppa 码 $\Gamma(L,g)$的校验矩阵为

$$\boldsymbol{H}=\begin{bmatrix} \xi_1 & \xi_2 & \cdots & \xi_n \\ \xi_1\alpha_1 & \xi_2\alpha_2 & \cdots & \xi_n\alpha_n \\ \vdots & \vdots & \ddots & \vdots \\ \xi_1\alpha_1^{t-1} & \xi_2\alpha_2^{t-1} & \cdots & \xi_n\alpha_n^{t-1} \end{bmatrix}$$

其次，考虑广义 Reed-Solomon 码 $\mathrm{GRS}_{n,k}(\boldsymbol{\alpha},\boldsymbol{\xi})$，$\boldsymbol{\alpha}=(\alpha_1,\alpha_2,\cdots,\alpha_n)$，$\boldsymbol{\xi}=(\xi_1,\xi_2,\cdots,\xi_n)$，即 $C=\{(\xi_1 f(\alpha_1),\xi_2 f(\alpha_2),\cdots,\xi_n f(\alpha_n))\mid f(x)\in F_{q^m}[x]_k\}$ 的生成矩阵。在 $\mathrm{GRS}_{n,k}(\boldsymbol{\alpha},\boldsymbol{\xi})$ 中，取 $k=t$，且设 $f(x)=\sum_{i=0}^{t-1}f_i x^i$。于是 C 中的码字可以写成

$$\left(\xi_1\sum_{i=0}^{t-1}f_i\alpha_1^i,\xi_2\sum_{i=0}^{t-1}f_i\alpha_2^i,\cdots,\xi_n\sum_{i=0}^{t-1}f_i\alpha_n^i\right)$$

$$=(f_0,f_1,\cdots,f_{n-1})\begin{bmatrix} \xi_1 & \xi_2 & \cdots & \xi_n \\ \xi_1\alpha_1 & \xi_2\alpha_2 & \cdots & \xi_n\alpha_n \\ \vdots & \vdots & \ddots & \vdots \\ \xi_1\alpha_1^{t-1} & \xi_2\alpha_2^{t-1} & \cdots & \xi_n\alpha_n^{t-1} \end{bmatrix}$$

这表明，Goppa 码 $\Gamma(L,g)$的校验矩阵正好为 $\mathrm{GRS}_{n,t}(\boldsymbol{\alpha},\boldsymbol{\xi})$的生成矩阵。因此，Goppa 码 $\Gamma(L,g)$是一个广义 Reed-Solomon 码的对偶码的子域($F_q\subset F_{q^m}$)子码。由定理 11.3.6 可知，Goppa 码 $\Gamma(L,g)$是一个广义 Reed-Solomon 码的子域($F_q\subset F_{q^m}$)子码。

特别地，当 $m=1$ 时，Goppa 码 $\Gamma(L,g)$就是一个广义 Reed-Solomon 码，即为 MDS 码。

下面的定理 11.3.8 说明了 Goppa 码 $\Gamma(L,g)$的基本参数。

定理 11.3.8　Goppa 码 $\Gamma(L,g)$的码长 $n=|L|$，维数 $k\geqslant n-mt$，极小距离 $d\geqslant$

$t+1$，其中，t 是 Goppa 多项式 $g(x)$ 的次数。

证明：由定理 11.3.7 的证明过程可知，Goppa 码 $\Gamma(L,g)$ 有以下形式的校验矩阵

$$\boldsymbol{H}=\begin{bmatrix}\xi_1 & \xi_2 & \cdots & \xi_n\\ \xi_1\alpha_1 & \xi_2\alpha_2 & \cdots & \xi_n\alpha_n\\ \vdots & \vdots & \ddots & \vdots\\ \xi_1\alpha_1^{t-1} & \xi_2\alpha_2^{t-1} & \cdots & \xi_n\alpha_n^{t-1}\end{bmatrix}$$

$$=\begin{pmatrix}1 & 1 & \cdots & 1\\ \alpha_1 & \alpha_2 & \cdots & \alpha_n\\ \alpha_1^2 & \alpha_2^2 & \cdots & \alpha_n^2\\ \vdots & \vdots & \ddots & \vdots\\ \alpha_1^{t-1} & \alpha_2^{t-1} & \cdots & \alpha_n^{t-1}\end{pmatrix}\begin{pmatrix}g(\alpha_1)^{-1} & 0 & 0 & 0 & 0\\ 0 & g(\alpha_2)^{-1} & 0 & 0 & 0\\ 0 & 0 & g(\alpha_3)^{-1} & 0 & 0\\ 0 & 0 & 0 & \ddots & 0\\ 0 & 0 & 0 & 0 & g(\alpha_n)^{-1}\end{pmatrix}$$

$\boldsymbol{H}$ 中的每个元素属于 F_{q^m}，可将这些元素用一组固定的基表示为 F_q 上的长为 m 的列向量(其元素属于 F_q)，从而 $\boldsymbol{H}$ 可以看作是 F_q 上的 $mt\times n$ 矩阵，它在 F_q 上的秩不大于 mt，于是 $k\geqslant n-mt$。由于 $\boldsymbol{H}$ 的任意列构成的行列式都是 Vandermonde 行列式且由已知条件可知均不为 0，所以由推论 11.2.1 得，$d\geqslant t+1$。

特别地，当 $q=2$ 时，即二元 Goppa 码，可证明其极小距离 d 在 $g(x)$(在 F_2 的扩域中)没有重根的条件下不小于 $2t+1$，即 $d\geqslant 2t+1$。对这一结果的证明感兴趣的读者可参阅文献[1]。

例 11.3.3 令 $g(x)=x^2+x+1$，$L=F_{2^3}=\{0,1,\alpha,\alpha^2,\cdots,\alpha^6\}$，$\alpha\in F_{2^3}$ 是本原元，满足 $\alpha^3+\alpha+1=0$。直接可验证 $g(\beta)\neq 0$，$\beta\in L=F_{2^3}$，所以 $g(x)$ 在 F_{2^3} 上不可约，由此可以得到一个不可约二元 Goppa 码。其参数为 $[n,k,d]=[n=2^3=8,k\geqslant 8-3\cdot 2=2,d\geqslant 2\deg g+1=5]$。由定理 11.3.7 的证明过程可得该码的校验矩阵为

$$\boldsymbol{H}=\begin{bmatrix}g^{-1}(0) & g^{-1}(1) & g^{-1}(\alpha) & \cdots & g^{-1}(\alpha^6)\\ 0g^{-1}(0) & g^{-1}(1) & \alpha g^{-1}(\alpha) & \cdots & \alpha^6 g^{-1}(\alpha^6)\end{bmatrix}$$

$$=\begin{bmatrix}1 & 1 & \alpha^2 & \alpha^4 & \alpha^2 & \alpha & \alpha & \alpha^4\\ 0 & 1 & \alpha^3 & \alpha^6 & \alpha^5 & \alpha^5 & \alpha^6 & \alpha^3\end{bmatrix}=\begin{bmatrix}1&1&0&0&0&0&0&0\\0&0&0&1&0&1&1&1\\0&0&1&1&1&0&0&1\\0&1&1&1&1&1&1&1\\0&0&1&0&1&1&0&1\\0&0&0&1&1&1&1&0\end{bmatrix}$$

11.3.4 二元 Reed-Muller 码

Reed-Muller 码是由 Muller 和 Reed 于 1954 年提出的一类重要的线性码，它不仅在纠错码中有着重要的地位，而且在密码学的研究中尤其是密码布尔函数的研究中具有重要的作用。为了简单起见，本节只讨论二元 Reed-Muller 码。这类码有很多等价的描述，这里用布尔函数的方式描述，以便于理解和进行应用举例。有关布尔

函数的概念和性质参见 10.1 节。

定义 11.3.4　设 m 是正整数，令 $B_m=\{f(x)\mid f(x):F_2^m\to F_2\}$，即 F_2 上全体 m 元布尔函数的集合，$n=2^m$，r 是整数且 $0\leqslant r\leqslant m$，r 阶二元 Reed-Muller 码定义为

$$C=\{\boldsymbol{f}\in F_2^n\mid f\in B_m,\deg f\leqslant r\}$$

简记为 RM(r,m)。

下面两个定理给出了 RM(r,m)的一些基本性质。

定理 11.3.9　设 m 是正整数，r 是整数且 $0\leqslant r\leqslant m$，则二元 RM(r,m) 是一个参数为$[n,k,d]=\left[2^m,\sum_{t=0}^{r}\binom{m}{t},2^{m-r}\right]$的线性码。

证明：因为 B_m 中所有满足 $\deg f\leqslant r$ 的 m 元布尔函数 $f(x)$构成 F_2 上的一个向量空间，由此可知，RM(r,m)是 F_2^n 的一个向量子空间，即 RM(r,m)是线性码。显然，码长为 $n=2^m$。

又因为次数不大于 r 的所有单项式所对应的码字

$$\{\boldsymbol{f}\in F_2^n\mid f=x_{i_1}x_{i_2}\cdots x_{i_t},0\leqslant t\leqslant r,1\leqslant i_1<i_2<\cdots<i_t\leqslant m\}$$

构成线性码 RM(r,m)在 F_2 上的一组基，因此 RM(r,m)的维数为

$$k=\binom{m}{0}+\binom{m}{1}+\cdots+\binom{m}{r}=\sum_{t=0}^{r}\binom{m}{t}$$

由引理 10.1.3 可知，RM(r,m)的极小距离 $d\geqslant 2^{m-r}$。易知 $f(x)=x_1x_2\cdots x_r$ 的重量是 2^{m-r}，因此，所对应的 $\boldsymbol{f}\in$RM(r,m)且 $W_{\mathrm{H}}(\boldsymbol{f})=2^{m-r}$，所以，$d=2^{m-r}$。

定理 11.3.10　设 m 是正整数，r 是整数且 $0\leqslant r\leqslant m-1$，则 RM$(r,m)^{\perp}=$ RM$(m-r-1,m)$。

证明：设 $\boldsymbol{f}\in$ RM(r,m)，$\boldsymbol{g}\in$ RM$(m-r-1,m)$，则 $\boldsymbol{f}$ 对应的布尔函数 f 与 $\boldsymbol{g}$ 对应的布尔函数 g 的乘积 fg 是一个次数不大于 $m-1$ 的 m 元布尔函数。于是由引理 10.1.2 可知，$W_{\mathrm{H}}(fg)$是偶数。因此，$\boldsymbol{f}\cdot\boldsymbol{g}=\sum_{a\in F_2^m}f(a)g(a)=W_{\mathrm{H}}(fg)=0(\mathrm{mod}\ 2)$。这表明 RM$(r,m)$ 中的码字与 RM$(m-r-1,m)$ 中的码子均正交。所以 RM$(r,m)^{\perp}\supseteq$ RM$(m-r-1,m)$。又由定理 11.3.9 可知，RM(r,m) 的维数是 $\sum_{t=0}^{r}\binom{m}{t}$，RM$(m-r-1,m)$ 的维数是 $\sum_{t=0}^{m-r-1}\binom{m}{t}$。由定理 11.2.3 可知，RM $(r,m)^{\perp}$ 的维数是 $2^m-\sum_{t=0}^{r}\binom{m}{t}$。而由组合论知识可知，$\sum_{t=0}^{r}\binom{m}{t}+\sum_{t=0}^{m-r-1}\binom{m}{t}=2^m$。所以 RM $(r,m)^{\perp}$ 的维数与 RM$(m-r-1,m)$ 的维数相等，于是，RM$(r,m)^{\perp}=$ RM$(m-r-1,m)$。

例 11.3.4　RM$(0,m)=\{(0,0,\cdots,0),(1,1,\cdots,1)\}\subseteq F_2^n$ 就是一个长为 $n=2^m$ 的二元重复码，即参数为$[n=2^m,1,2^m]$的线性码，其生成矩阵为 $\boldsymbol{G}=(1,1,\cdots,1)$。由定理 11.3.9 可知，$\boldsymbol{G}=(1,1,\cdots,1)$是 RM$(m-1,m)$的校验矩阵，即

$$c=(c_1,c_2,\cdots,c_n)\in \mathrm{RM}(m-1,m)\quad 当且仅当\quad c_1+c_2+\cdots+c_n=0$$

这表明 RM$(m-1,m)$是一个长为 n 的二元奇偶校验码，其参数为$[n=2^m,2^m-1,2]$。

11.4 一些典型的译码方法

一般而言,编码是比较容易实现的,难点在于译码,一般线性码的译码问题都是一个 NP 完全问题。编码理论的一个中心任务就是设计有效的译码方法。译码方法大体上可以分为两类:一类是可用于任意码的一般译码方法,如 11.4.1 小节、11.4.2 小节和 11.4.3 小节介绍的译码方法;另一类是用于特定的码或码类的专用译码方法,如 11.4.4 小节和 11.4.5 小节介绍的译码方法。通常,后者比前者更为快捷和简便。值得一提的是,11.4.3 小节介绍的校验子(也称伴随式)译码方法可以作为衡量其他译码方法的标准。换言之,在采用一种新的译码方法之前,最好先和校验子译码方法进行比较。

11.4.1 极小距离译码

设 $C\subseteq A^n$ 是定义 11.1.2 中所定义的码。利用 Hamming 距离的概念,可以按照以下方法译码:当接收者接收到字 $\boldsymbol{r}$,首先找到码字 $\boldsymbol{x}$ 使得 $d_H(\boldsymbol{x},\boldsymbol{r})$ 尽可能地小,就将 $\boldsymbol{r}$ 译为 $\boldsymbol{x}$。注意到给定一个接收到的字 $\boldsymbol{r}$,可能存在多于一个的有效码字与 $\boldsymbol{r}$ 的 Hamming 距离同样小。在这种情况下,不能保证完全正确地纠错,而只是选取其中一个最接近的码字。这种译码方法就被称为**极小距离译码**。

11.4.2 大数逻辑译码

大数逻辑译码方法不仅是纠错码中的重要译码方法,可用于很多线性码的译码,而且也是密码分析尤其是序列密码分析中的重要工具。

定义 11.4.1 一组校验方程 $\boldsymbol{x}\cdot\boldsymbol{y}^{(v)}=0(v=1,2,\cdots,r)$ 关于位置 i(对码 C,$\boldsymbol{y}^{(v)}\in C^{\perp}$)被称为是正交的,如果:① $y_i^{(v)}=1(v=1,2,\cdots,r)$,$y_i^{(v)}$ 表示 $\boldsymbol{y}^{(v)}$ 的第 i 个分量;② 若 $j\neq i$,则至多有一个 v 使得 $y_j^{(v)}\neq 0$。

现在假定 $\boldsymbol{x}$ 是一个包含 t 个错误的接收字,$t\leqslant\frac{1}{2}r$,则有

(1) 若 x_i 是正确的,则至多有 t 个 v 使得 $\boldsymbol{x}\cdot\boldsymbol{y}^{(v)}\neq 0$;

(2) 若 x_i 是不正确的,则至少有 $r-(t-1)$ 个 v 使得 $\boldsymbol{x}\cdot\boldsymbol{y}^{(v)}\neq 0$。

因为 $r-(t-1)>t$,所以可以通过 $\boldsymbol{x}\cdot\boldsymbol{y}^{(v)}$ 的多数值(0 或不是 0)来确定 x_i 正确还是不正确。在二元码的情况下,我们能纠正错误。如果对每个 i 有这样的正交校验集,则能逐一纠正不同位置的错误。

例 11.4.1 考虑[7,4] 二元 Hamming 码的对偶码。下列 3 个校验方程关于位置 1 是正交的:

$$x_1+x_2+x_3=0$$
$$x_1+x_4+x_5=0$$
$$x_1+x_6+x_7=0$$

如果 $\boldsymbol{x}$ 包含 1 个错误,则若 x_1 是不正确的,则 3 个校验方程产生 3 个 1;若 x_1 是正确的,则 3 个校验方程产生 2 个 0、1 个 1。

11.4.3 校验子译码

假设 Alice 发送给 Bob 的码字是 $\boldsymbol{c}$，Bob 接收到的字是 $\boldsymbol{r}$，可将 $\boldsymbol{r}$ 写为 $\boldsymbol{r}=\boldsymbol{c}+\boldsymbol{e}$，其中 $\boldsymbol{e}\in F_q^n$，$\boldsymbol{e}$ 表示传输中发生的错误，称 $\boldsymbol{e}$ 为错误向量。Bob 知道 $\boldsymbol{r}$，现在的问题是他如何找出 $\boldsymbol{c}$ 的值。显然只要能找到 $\boldsymbol{e}$ 即可找到 $\boldsymbol{c}$。本节介绍一种找到 $\boldsymbol{e}$ 的方法。

定义 11.4.2 设 $C\subseteq F_q^n$ 是一个线性码，$\boldsymbol{x}\in F_q^n$。定义陪集 $\boldsymbol{x}+C$ 为

$$\boldsymbol{x}+C=\{\boldsymbol{x}+\boldsymbol{c}\mid \boldsymbol{c}\in C\}$$

关于陪集，可用定义直接证明下述结论。

定理 11.4.1 令 $C\subseteq F_q^n$ 是一个线性码，则

(1) 若 $\boldsymbol{x}\in \boldsymbol{y}+C$，则 $\boldsymbol{x}+C=\boldsymbol{y}+C$；

(2) 对每一对 $\boldsymbol{x},\boldsymbol{y}\in F_q^n$，或者 $\boldsymbol{x}+C=\boldsymbol{y}+C$ 成立或者 $\boldsymbol{x}+C\cap \boldsymbol{y}+C=\varnothing$ 成立。

假定数据传输过程中发生的错误不太多，即 $W_{\mathrm{H}}(\boldsymbol{e})$ 较小，Bob 计算集合 $\boldsymbol{x}+C$ 中所有元素的重量，具有最小重量的元素若唯一，记为 $\boldsymbol{e}$。

例 11.4.2 令 $C\subseteq F_2^4$ 定义为

$$C=\{(0000),(1100),(0011),(1111)\}$$

若 $\boldsymbol{x}=(1010)$，则有

$$\boldsymbol{x}+C=\{(1010),(0110),(1001),(0101)\}$$

现在定义一个字的校验子（又称伴随式），校验子可以帮助我们标识线性码的陪集。

定义 11.4.3 若 $C\subseteq F_q^n$ 是一个参数为 $[n,k]$ 的线性码，$\boldsymbol{H}$ 是 C 的一个校验矩阵。令 $\boldsymbol{r}\in F_q^n$，则 $\boldsymbol{s}=\boldsymbol{H}\boldsymbol{r}^{\mathrm{T}}\in F_q^{n-k}$ 称为 $\boldsymbol{r}$ 的校验子，也称为 $\boldsymbol{r}$ 的伴随式。

例 11.4.3 参数为 $[7,4]$ 的二元 Hamming 码的一个校验矩阵是

$$\boldsymbol{H}=\begin{pmatrix}1&1&1&0&1&0&0\\1&1&0&1&0&1&0\\0&1&1&1&0&0&1\end{pmatrix}$$

则 (1110000) 的校验子为

$$\begin{pmatrix}1&1&1&0&1&0&0\\1&1&0&1&0&1&0\\0&1&1&1&0&0&1\end{pmatrix}\begin{pmatrix}1\\1\\1\\0\\0\\0\\0\end{pmatrix}=\begin{pmatrix}1\\0\\0\end{pmatrix}$$

我们知道，$\boldsymbol{r}\in C$ 当且仅当 $\boldsymbol{H}\boldsymbol{r}^{\mathrm{T}}=\boldsymbol{0}$。也就是说，$\boldsymbol{r}\in C$ 当且仅当它的校验子是 $\boldsymbol{0}$。由此可以看出，Bob 可以用一个更好的方法进行纠错，他可以不用拿接收到的字 $\boldsymbol{r}$ 与所有的码字比较，而是直接计算它的校验子，校验子为 $\boldsymbol{0}$ 当且仅当没有错误发生。下面将用校验子的概念进行纠错。

可以把 Bob 接收的字写为 $\boldsymbol{r}=\boldsymbol{c}+\boldsymbol{e}$，其中 $\boldsymbol{c}$ 是 Alice 发送的码字。显然，$\boldsymbol{r}$ 与 $\boldsymbol{e}$ 有

相同的校验子当且仅当 $\boldsymbol{Hr}^{\mathrm{T}}=\boldsymbol{He}^{\mathrm{T}}$，当且仅当 $\boldsymbol{H}(\boldsymbol{r}-\boldsymbol{e})^{\mathrm{T}}=\boldsymbol{0}$，当且仅当 $\boldsymbol{r}-\boldsymbol{e}\in C$。但是 Bob 知道 $\boldsymbol{r}-\boldsymbol{e}=\boldsymbol{c}\in C$，这说明 $\boldsymbol{r}$ 必须与 $\boldsymbol{e}$ 有相同的校验子。所以 Bob 想找到一个具有较小重量的字 $\boldsymbol{r}$ 使得 $\boldsymbol{r}$ 与 $\boldsymbol{e}$ 有相同的校验子。

关于校验子，容易证明下述结论。

定理 11.4.2 设 $C\subseteq F_q^n$ 是一个线性码，则 $\boldsymbol{x},\boldsymbol{y}\in F_q^n$ 有相同的校验子当且仅当 $\boldsymbol{x}\in\boldsymbol{y}+C$。

定理 11.4.2 说明，两个字有相同的校验子当且仅当它们在同一个陪集中，而且由定理 11.4.1 可知，陪集基于校验子把集合 F_q^n 分解为互不相交的部分。

如果 Bob 知道 $\boldsymbol{r}$ 与 $\boldsymbol{e}$ 有相同的校验子，那么他也知道所有与 $\boldsymbol{r}$ 有相同的校验子的元素都在陪集 $\boldsymbol{r}+C$ 中。所以他观察 $\boldsymbol{r}+C$ 中的所有元素，并从中选取具有最小重量的 $\boldsymbol{e}$。

定义 11.4.4 在一个陪集中，具有最小重量的元素称为陪集首项。

注意：对一个特定的陪集而言，陪集首项可能不唯一。

可以把校验子译码算法概述如下。

(1) Bob 接收到字 $\boldsymbol{r}$。

(2) 他计算 $\boldsymbol{r}$ 的校验子 $\boldsymbol{s}=\boldsymbol{H}\boldsymbol{r}^{\mathrm{T}}$。

(3) 若 $\boldsymbol{s}=\boldsymbol{0}$，则没有错误发生。

(4) 若 $\boldsymbol{s}\neq\boldsymbol{0}$，则 Bob 观察所有元素都有校验子 $\boldsymbol{s}$ 的陪集，找出陪集首项并假定为 $\boldsymbol{e}$。

(5) 他计算 $\boldsymbol{c}=\boldsymbol{r}-\boldsymbol{e}$ 进行纠错。

这种方法的优点在于 Bob 能够在 Alice 开始发送消息之前计算并储存一个校验子和陪集首项的表，则当他收到消息之后，纠错是很快的。

例 11.4.4 参数为[7,4]的二元 Hamming 码的一个校验矩阵是

$$\boldsymbol{H}=\begin{pmatrix}1&1&1&0&1&0&0\\1&1&0&1&0&1&0\\0&1&1&1&0&0&1\end{pmatrix}$$

如果 Alice 和 Bob 约定使用这个码，则 Bob 在 Alice 发送消息之前计算如下。

校验子	陪集首项	校验子	陪集首项
$\begin{pmatrix}0\\0\\0\end{pmatrix}$	$(0\ 0\ 0\ 0\ 0\ 0\ 0)$	$\begin{pmatrix}0\\1\\1\end{pmatrix}$	$(0\ 0\ 0\ 1\ 0\ 0\ 0)$
$\begin{pmatrix}0\\0\\1\end{pmatrix}$	$(0\ 0\ 0\ 0\ 0\ 0\ 1)$	$\begin{pmatrix}1\\0\\1\end{pmatrix}$	$(0\ 0\ 1\ 0\ 0\ 0\ 0)$
$\begin{pmatrix}0\\1\\0\end{pmatrix}$	$(0\ 0\ 0\ 0\ 0\ 1\ 0)$	$\begin{pmatrix}1\\1\\0\end{pmatrix}$	$(0\ 0\ 1\ 0\ 0\ 0\ 0)$
$\begin{pmatrix}1\\0\\0\end{pmatrix}$	$(0\ 0\ 0\ 0\ 1\ 0\ 0)$	$\begin{pmatrix}1\\1\\1\end{pmatrix}$	$(0\ 1\ 0\ 0\ 0\ 0\ 0)$

假设 Bob 接收到 $\boldsymbol{r}=(1011100)$，则计算 $\boldsymbol{r}$ 的校验子得到 $\boldsymbol{s}=\boldsymbol{H}\boldsymbol{r}^{\mathrm{T}}=\begin{pmatrix}1\\0\\0\end{pmatrix}$。

他接着查表找向量 $\begin{pmatrix}1\\0\\0\end{pmatrix}$ 的陪集首项和校验子。由于他接收到的字为(0000100)，他知道(1011100)－(0000100)＝(1011000)是一个有效的码字，所以 Bob 把(1011100)纠正为(1011000)。

11.4.4　BCH 码的译码

设 C 是定义 11.3.1 中所定义的 BCH 码。则 C 的设计距离为 $\delta=2t+1$，所以可纠正不大于 t 个错误。设 $c(x)=\sum_{i=0}^{n-1}c_ix^i\in C$，错误多项式是 $e(x)=\sum_{i=0}^{n-1}e_ix^i\in F_q[x]$，$W_{\mathrm{H}}(e(x))\leqslant t$。定义错位集为 $M=\{i\mid e_i\neq 0,0\leqslant i\leqslant n-1\}$，$l=|M|=W_{\mathrm{H}}(e(x))\leqslant t$。

译码的目的就是从收到的 $r(x)=c(x)+e(x)=\sum_{i=0}^{n-1}r_ix^i\in F_q[x]$ 计算出错位集 M 和错值 $e_i(i\in M)$。为了求出错位集 M，只需求出 $\alpha^i(i\in M)$，定义错位多项式为

$$\sigma(x)=\prod_{i\in M}(1-\alpha^ix),\quad \deg\sigma(x)=l\leqslant t$$

定义错值多项式为

$$\omega(x)=\sigma(x)\sum_{i\in M}\frac{e_i\alpha^ix}{1-\alpha^ix}=\sum_{i\in M}e_i\alpha^ix\prod_{j\in M\setminus\{i\}}(1-\alpha^jx)$$

若能求出多项式 $\sigma(x)$ 和 $\omega(x)$，则由 $\sigma(x)$ 可确定错位($i\in M$ 当且仅当 $\sigma(\alpha^{-i})=0$)。然后对 $i\in M$ 可求出错值

$$e_i=-\omega(\alpha^{-i})\alpha^i/\sigma'(\alpha^{-i})$$

其中 $\sigma'(x)$ 表示多项式 $\sigma(x)$ 的形式微分。

因为 $\frac{\omega(x)}{\sigma(x)}=\sum_{i\in M}\frac{e_i\alpha^ix}{1-\alpha^ix}=\sum_{i\in M}e_i\sum_{j=1}^{\infty}(\alpha^ix)^j=\sum_{j=1}^{\infty}x^j\sum_{i\in M}e_i\alpha^{ij}=\sum_{j=1}^{\infty}e(\alpha^j)x^j$，所以当 $1\leqslant j\leqslant 2t$ 时，由 $c(\alpha^j)=0$ 可知 $e(\alpha^j)=r(\alpha^j)$。而这些 $e(\alpha^j)(1\leqslant j\leqslant 2t)$ 可由收到的 $r(x)$ 计算出来，因此接收者知道 $\sum_{j=1}^{\infty}e(\alpha^j)x^j$ 的前 $2t$ 个系数。所以接收者知道 $\frac{\omega(x)}{\sigma(x)}\bmod x^{2t+1}$。

假定接收者找到了多项式 $\omega(x)$ 和次数尽可能小的多项式 $\sigma(x)$，使得 $\deg\omega(x)\leqslant\deg\sigma(x)$，$\frac{\omega(x)}{\sigma(x)}\equiv\sum_{j=1}^{2t}e(\alpha^j)x^j \bmod x^{2t+1}$。

令 $s_j=e(\alpha^j)(1\leqslant j\leqslant 2t)$，$\sigma(x)=\prod_{i\in M}(1-\alpha^ix)=\sum_{i=0}^{l}\sigma_ix^i$，$\sigma_0=1$，$l=|M|$，则

$$\omega(x)\equiv\sigma(x)\sum_{j=1}^{2t}s_jx^j=\sum_k\Big(\sum_{i+j=k}s_j\sigma_i\Big)x^k \bmod x^{2t+1}$$

因为 $\deg\omega(x)\leqslant l$，所以

$$\sum_{i+j=k} s_j\sigma_i = 0,\quad l+1\leqslant k\leqslant 2t$$

由此给出了关于未知量 $\sigma_1,\sigma_2,\cdots,\sigma_l$ 的 $2t-l$ 个方程。设 $\tilde{\sigma}(x)=\sum_{i=0}^{l}\tilde{\sigma}_i x^i(\tilde{\sigma}_0=1)$ 是由上述方程组给出的次数最低的多项式，这样的多项式一定存在，因为 $\sigma(x)$ 即是。所以对 $l+1\leqslant k\leqslant 2t$，有

$$\begin{aligned} 0 &= \sum_i s_{k-i}\tilde{\sigma}_i = \sum_i r(\alpha^{k-i})\tilde{\sigma}_i = \sum_i e(\alpha^{k-i})\tilde{\sigma}_i \\ &= \sum_{i\in M}\sum_j e_j\alpha^{(k-i)j}\tilde{\sigma}_i = \sum_{j\in M} e_j\alpha^{kj}\tilde{\sigma}(\alpha^{-j}) \end{aligned}$$

令 $x_j=e_j\tilde{\sigma}(\alpha^{-j})$，则有

$$\sum_j \alpha^{kj}x_j = 0,\quad l+1\leqslant k\leqslant 2t$$

可将上面 $2t-l\geqslant l$ 个方程视作一个关于未知量 $x_1,x_2,\cdots,x_l$ 的方程组，注意到由前 l 个方程组成的方程组的系数矩阵是一个 Vandermonde 矩阵，易知其行列式不为零，所以 $x_1=x_2=\cdots=x_l=0$，即 $x_j=e_j\tilde{\sigma}(\alpha^{-j})=0$，$1\leqslant j\leqslant l$。因此，$j\in M$ 当且仅当 $\tilde{\sigma}(\alpha^{-j})=0$。由 $\sigma(x)$ 的定义可知，$\sigma(x)\mid\tilde{\sigma}(x)$。因此由 $\tilde{\sigma}(x)$ 的极小性可知，$\sigma(x)=\tilde{\sigma}(x)$。

现在的问题是接收者如何快速地找到 $\sigma(x)$？由上述讨论可知，接收者总是可以通过解方程组来找到 $\sigma(x)$，但可用更有效的算法——Berlekamp-massey 算法来找到 $\sigma(x)$。接收者首先计算长为 $2t$ 的序列 $s_j=r(\alpha^j)=e(\alpha^j)(1\leqslant j\leqslant 2t)$，也称校验子或伴随式。其次利用 Berlekamp-massey 算法找到生成该序列的最短 LFSR 的连接多项式 $\tilde{\sigma}(x)$，$\tilde{\sigma}(x)$ 即为所求，即 $\sigma(x)=\tilde{\sigma}(x)$。记以 $\sigma(x)$ 为连接多项式的 LFSR 生成的序列为 $\{s_j\}_{j=1}^{\infty}$，其中 $s_j=r(\alpha^j)=e(\alpha^j)(1\leqslant j\leqslant 2t)$。

综上所述，BCH 码的译码算法可以描述为：

设发送的码字是 $c(x)=\sum_{i=0}^{n-1}c_i x^i\in C$，错误是 $e(x)=\sum_{i=0}^{n-1}e_i x^i\in F_q[x]$，$W_{\mathrm{H}}(e(x))\leqslant t$，收到的字是 $r(x)=c(x)+e(x)=\sum_{i=0}^{n-1}r_i x^i\in F_q[x]$。

第 1 步，计算校验子 $s_j=r(\alpha^j)$，$1\leqslant j\leqslant 2t$。

第 2 步，求 $\sigma(x)$ 和 $\omega(x)$，利用 Berlekamp-massey 算法找到生成序列 $s_j=r(\alpha^j)$ $(1\leqslant j\leqslant 2t)$ 的最短 LFSR 的连接多项式 $\sigma(x)$，$\deg\sigma(x)=l\leqslant t$，记 $\{s_j\}_{j=1}^{\infty}$ 是以 $\sigma(x)$ 为连接多项式的 LFSR 生成的序列，令 $\omega(x)=\sigma(x)\sum_{j=1}^{\infty}s_j x^j$。

第 3 步，求错位，求出 $\sigma(x)$ 的全部根，即 α^{-i_k}，$0\leqslant i_k\leqslant n-1$，$k=1,2,\cdots,l$，则错位集为 $M=\{i_1,i_2,\cdots,i_l\}$。

第 4 步，求错值，错位 $i\in M$ 处的错值是 $e_i=-\omega(\alpha^{-i})\alpha^i/\sigma'(\alpha^{-i})$，于是 $e(x)=\sum_{i\in M}e_i x^i$，则正确的码字是 $c(x)=r(x)-e(x)$。

11.4.5　Goppa 码的译码

Goppa 码可用类似于 11.4.4 小节所介绍的方法进行译码，下面来讨论这个问题。设 $\Gamma(L,g)$ 是定义 11.3.3 中所定义的 Goppa 码。令 $\boldsymbol{c}=(c_1,c_2,\cdots,c_n)\in\Gamma(L,g)$，假定接收者收到的字是 $\boldsymbol{r}=(r_1,r_2,\cdots,r_n)$。错误向量为 $\boldsymbol{e}=(e_1,e_2,\cdots,e_n)=\boldsymbol{r}-\boldsymbol{c}$。定义错位集为 $M=\{i\mid e_i\neq 0\}$，$|M|=e<t/2$，$\deg g=t$。

定义伴随式多项式为

$$S(x)\equiv\sum_{i=1}^{n}\frac{e_i}{x-\alpha_i}\bmod g(x)$$

显然伴随式多项式 $S(x)$ 可由 $\boldsymbol{r}$ 来计算，这是因为 $\sum_{i=1}^{n}\frac{c_i}{x-\alpha_i}\equiv 0 \bmod g(x)$，所以

$$S(x)\equiv\sum_{i=1}^{n}\frac{e_i}{x-\alpha_i}\equiv\sum_{i=1}^{n}\frac{r_i-c_i}{x-\alpha_i}\equiv\sum_{i=1}^{n}\frac{r_i}{x-\alpha_i}\bmod g(x)$$

定义错位多项式为

$$\sigma(x)=\prod_{i\in M}(x-\alpha_i)$$

定义错值多项式为

$$\omega(x)=\sigma(x)\sum_{i\in M}\frac{e_i}{x-\alpha_i}=\sum_{i\in M}e_i\prod_{j\in M\setminus\{i\}}(x-\alpha_i)$$

显然，$\gcd(\sigma(x),\omega(x))=1$，$\deg\sigma(x)=e$，$\deg\omega(x)<e$。

直接可验证 $S(x)$、$\sigma(x)$ 和 $\omega(x)$ 之间有以下关系

$$S(x)\sigma(x)\equiv\sum_{i=1}^{n}\frac{e_i}{x-\alpha_i}\prod_{i\in M}(x-\alpha_i)\equiv\omega(x)\bmod g(x)\tag{11.1}$$

现在假定有一个算法可以找到次数最低的首 1 非零多项式 $\sigma_1(x)$ 和次数最低的多项式 $\omega_1(x)$，使得

$$S(x)\sigma_1(x)\equiv\omega_1(x)\bmod g(x)\tag{11.2}$$

设 $\gcd(S(x),g(x))=d(x)$，则由式(11.1)和式(11.2)分别可得

$$S(x)\sigma(x)/d(x)\equiv\omega(x)/d(x)\bmod g(x)/d(x)$$

$$\omega_1(x)/d(x)\equiv S(x)\sigma_1(x)/d(x)\bmod g(x)/d(x)$$

将上面两式左右两边分别相乘并移项整理得

$$S(x)/d(x)[\sigma(x)\omega_1(x)-\sigma_1(x)\omega(x)]/d(x)\equiv 0 \bmod g(x)/d(x)$$

所以 $\sigma(x)\omega_1(x)-\sigma_1(x)\omega(x)\equiv 0 \bmod g(x)$，由于此式的左边的次数低于 $g(x)$ 的次数，所以 $\sigma(x)\omega_1(x)-\sigma_1(x)\omega(x)=0$，又 $\gcd(\sigma(x),\omega(x))=1$，因此，$\sigma(x)$ 是 $\sigma_1(x)$ 的因式，$\omega(x)$ 是 $\omega_1(x)$ 的因式，所以必须有 $\sigma(x)=\sigma_1(x)$，$\omega(x)=\omega_1(x)$。

由此可见，只要能找到 $\sigma(x)$ 和 $\omega(x)$ 就能找到 $\boldsymbol{e}$。事实上，Berlekamp-massey 算法或 Euclid 算法就是寻找 $\sigma_1(x)$ 的有效算法。

综上所述，Goppa 码的译码算法可以描述为：

设发送的码字是 $\boldsymbol{c}=(c_1,c_2,\cdots,c_n)\in\Gamma(L,g)$，错误是 $\boldsymbol{e}=(e_1,e_2,\cdots,e_n)$，接收者收到的字是 $\boldsymbol{r}=(r_1,r_2,\cdots,r_n)=\boldsymbol{c}+\boldsymbol{e}$。

第 1 步，计算伴随式多项式 $S(x) \equiv \sum_{i=1}^{n} \frac{r_i}{x-\alpha_i} \bmod g(x) = \sum_{i=1}^{t} s_i x^{i-1}$。

第 2 步，求 $\sigma(x)$ 和 $\omega(x)$，利用 Berlekamp-massey 算法找到生成序列 $s_j(1 \leqslant j \leqslant t)$ 的最短 LFSR 的连接多项式 $\sigma(x)$，$\deg\sigma(x) = e < t/2$，记 $\{s_j\}_{j=1}^{\infty}$ 是以 $\sigma(x)$ 为连接多项式的 LFSR 生成的序列，令 $\omega(x) = \sigma(x)\sum_{j=1}^{\infty} s_j x^j$。

第 3 步，求错位，求出 $\sigma(x)$ 的全部根，即 α_{i_k}，$0 \leqslant i_k \leqslant n-1$，$k=1,2,\cdots,e$ 则错位集为 $M=\{i_1,i_2,\cdots,i_e\}$。

第 4 步，求错值，错位 $i \in M$ 处的错值是 $e_i = \omega(\alpha_i)/\sigma'(\alpha_i)$，于是 $\boldsymbol{e}=(e_1,e_2,\cdots,e_n)$，则正确的码字是 $\boldsymbol{c}=\boldsymbol{r}-\boldsymbol{e}$。

11.5 应用举例

纠错码方法与技术在信息安全领域有着广泛而深入的应用，在以下 3 个方面取得了一批重要成果。一是将纠错码方法与技术用于研究密码算法和安全协议的组件的设计和分析，通过建立纠错码和密码学中的一些概念的等价性，将一些概念联系起来，如 Reed-Muller 码的覆盖半径与布尔函数的非线性度、函数的相关免疫阶与码的对偶距离、P 置换与 MDS 码，以推动两个学科的交叉研究。二是将纠错码方法与技术用于研究密码算法和安全协议的分析，在有些情况下，密文序列可视作一个含错明文序列，如基于 LFSR 的序列密码，所以，利用纠错码中的译码方法和技术进行密码分析是很自然的，往往也是很有效的。三是将纠错码方法与技术用于研究密码算法和安全协议的设计，包括基于纠错码的公钥加密算法、私钥加密算法，基于纠错码的数字签名方案、密钥管理协议、秘密共享协议、认证协议等。限于篇幅，这里只简要介绍两个基本的应用。

11.5.1 基于纠错码的公钥加密算法——McEliece 密码算法

McEliece 密码算法是 McEliece 于 1978 年提出的一种基于纠错码的公钥加密算法，它的安全性是基于一般线性码的译码问题，是一个困难性问题即 NP 完全问题。到目前为止，仍然认为该算法是安全的，但该算法存在着一些缺陷，诸如公开密钥庞大，数据扩展大。McEliece 密码算法的设计思想与传统的 Merkle-Hellman 密码算法的设计思想相似，译一般线性码是一个 NP 完全问题，但一些好码诸如 Goppa 码有多项式时间的译码算法。McEliece 的策略是将一个有多项式时间译码算法的 Goppa 码伪装成一个在没有辅助信息下难译的线性码。

下面就来描述 McEliece 密码算法。

设 $\boldsymbol{G}$ 是一个参数为$[n,k,d]$的二元不可约 Goppa 码的生成矩阵，$n=2^m$，$k=n-mt$，$d=2t+1$。设明文空间为 F_2^k，密文空间为 F_2^n。

密钥的生成过程如下：

随机地选取 F_2 上的 $k \times k$ 阶可逆矩阵 $\boldsymbol{S}$ 和 $n \times n$ 阶置换矩阵 $\boldsymbol{P}$，令 $\boldsymbol{G}'=\boldsymbol{SGP}$，将

$\boldsymbol{G}$、$\boldsymbol{S}$ 和 $\boldsymbol{P}$ 作为私钥进行保密，将 $\boldsymbol{G}'$ 作为公钥公开。

加密过程如下：

对任意一个明文 $m\in F_2^k$，对应的密文定义为 $c=m\boldsymbol{G}'+e$，这里 $e\in F_2^n$ 是一个重量不超过 t 的随机向量。

解密过程如下：

收方收到一个密文 c，计算 $c\boldsymbol{P}^{-1}=m\boldsymbol{SGPP}^{-1}+e\boldsymbol{P}^{-1}=m\boldsymbol{SG}+e'$，因为 $\boldsymbol{P}$ 是一个置换矩阵，所以 e 与 e' 有相同的重量。利用 Goppa 码的快速译码算法将其译码成 $m'=m\boldsymbol{S}$，密文 c 对应的明文为 $m=m'\boldsymbol{S}^{-1}$。

在该算法的实际的实现中，McEliece 建议选取以下参数：$m=10$，$t=50$，此时 $n=2^{10}=1024$，$k=n-mt=524$，$d=101$，Goppa 码是一个参数为$[1024,524,101]$的线性码。明文长度为 524bit，密文长度为 1024bit。公钥是一个 524×1024 阶的二元矩阵。

McEliece 密码算法的实现过程中主要涉及两个方面的内容，一是关于二元 Goppa 码的基本性质；二是 F_{2^m} 上的 t 次不可约多项式的选取与测试。关于第一个问题，如二元 Goppa 码的码长 $n=2^m$，维数 $k\geqslant n-mt$，最小距离 $d\geqslant 2t+1$，存在时间复杂度为 $O(nt)$的快速纠错译码算法，已在 11.4.5 小节做了介绍，也可参阅文献[1]。关于第二个问题，由有限域的知识可知，F_{2^m} 上的 t 次不可约多项式有 $O\left(\frac{2^{mt}}{t}\right)$个，这表明，对给定的 m，t，用随机取法能以$\frac{1}{t}$的概率获得 t 次不可约多项式，其不可约性有快速算法加以测试，参见文献[2]。

11.5.2　基于纠错码的数字签名方案——AW 数字签名方案

AW 数字签名方案是 Alabbadi 和 Wicker 于 1992 年在对 Xinmei 数字签名方案的安全性分析的基础上提出的一个改进方案，不少学者对此类方案进行了分析研究，已有多种分析结果。本节除了介绍该方案之外，还介绍了一种利用求矩阵广义逆实现伪造攻击的思想，证明了 AW 数字签名方案存在严重的安全漏洞，即任何人仅利用签名用户的公钥就可以构造出等价的签名私钥，从而成功地实现伪造签名。

1. AW 数字签名方案

假定通信双方是用户 A 和用户 B。首先用户 A 选择一个非线性函数 $f(x,y)$：$F_2^k\times F_2^n\rightarrow F_2^k$ 并把它公开。

用户 A 选择一个$[n,k,d\geqslant 2t+1]$二元 Goppa 码，以及其生成矩阵 $\boldsymbol{G}$ 和校验矩阵 $\boldsymbol{H}$。矩阵 $\boldsymbol{G}^*$ 满足 $\boldsymbol{GG}^*=I_k$，$\boldsymbol{P}$ 是 F_2 上一个满秩的 $n\times n$ 阶随机矩阵，$\boldsymbol{W}$ 是一个秩为 n 的 $n\times l$ 阶矩阵，这里 $l>n$，矩阵 $\boldsymbol{W}^*$ 满足 $\boldsymbol{WW}^*=I_n$。用户 A 的公钥为($\boldsymbol{G}'$，$\boldsymbol{H}'$，$\boldsymbol{W}^*$，$\boldsymbol{H}$，t')，其中 $\boldsymbol{G}'=\boldsymbol{P}^{-1}\boldsymbol{G}^*$，$\boldsymbol{H}'=\boldsymbol{P}^{-1}\boldsymbol{H}^{\mathrm{T}}$，$t'<t$。$A$ 保留其私钥 $\boldsymbol{G}$，$\boldsymbol{P}$，$\boldsymbol{G}^*$，$\boldsymbol{W}$。

产生签名的过程如下：

用户 A 随机地选取一个 Hamming 重量 $W_{\mathrm{H}}(\boldsymbol{E})\leqslant t$ 的向量 $\boldsymbol{E}\in F_2^n$ 和一个 Hamming 重量 $W_{\mathrm{H}}(\boldsymbol{E})\approx l/2$ 的向量 $\boldsymbol{Z}\in F_2^l$，对消息 M 的签名是一个长度为 l 的二元

序列：

$$C = \{(\boldsymbol{E} + (f(M,\boldsymbol{E}) + \boldsymbol{EG}^*)\boldsymbol{G})\boldsymbol{P} + \boldsymbol{ZW}^*\}\boldsymbol{W} + \boldsymbol{Z} \tag{11.3}$$

用户 A 把(M,C)发送给用户 B。

验证签名的过程如下：

当用户 B 接收到(M,C)后，首先计算

$$\begin{aligned} V = C\boldsymbol{W}^* &= \{(\boldsymbol{E} + (f(M,\boldsymbol{E}) + \boldsymbol{EG}^*)\boldsymbol{G})\boldsymbol{P} + \boldsymbol{ZW}^*\}\boldsymbol{WW}^* + \boldsymbol{ZW}^* \\ &= \{(\boldsymbol{E} + (f(M,\boldsymbol{E}) + \boldsymbol{EG}^*)\boldsymbol{G})\boldsymbol{P} + \boldsymbol{ZW}^*\} + \boldsymbol{ZW}^* \\ &= \boldsymbol{EP} + (f(M,\boldsymbol{E}) + \boldsymbol{EG}^*)\boldsymbol{GP} \end{aligned}$$

这里用到了 $\boldsymbol{WW}^* = I_n$。然后利用 A 的公钥 $\boldsymbol{H}' = \boldsymbol{P}^{-1}\boldsymbol{H}^{\mathrm{T}}$ 计算伴随式

$$\boldsymbol{VH}' = \{\boldsymbol{EP} + (f(M,\boldsymbol{E}) + \boldsymbol{EG}^*)\boldsymbol{GP}\}\boldsymbol{P}^{-1}\boldsymbol{H}^{\mathrm{T}} = \boldsymbol{EH}^{\mathrm{T}}$$

因为 $\boldsymbol{GH}^{\mathrm{T}} = 0$，从而利用 Berlekamp-Massey 算法译码即可得到 $\boldsymbol{E}$。另一方面

$$\begin{aligned} \boldsymbol{VG}' &= \{\boldsymbol{EP} + (f(M,\boldsymbol{E}) + \boldsymbol{EG}^*)\boldsymbol{GP}\}\boldsymbol{P}^{-1}\boldsymbol{G}^* \\ &= \boldsymbol{EG}^* + f(M,\boldsymbol{E}) + \boldsymbol{EG}^* = f(M,\boldsymbol{E}) \end{aligned} \tag{11.4}$$

用户 B 使用收到的消息 M 和译码得到的 $\boldsymbol{E}$ 作为输入，计算函数值 $f(M,\boldsymbol{E})$，若该值与式(11.4)右端的计算结果相等，则签名有效，否则拒绝该签名。

2. *AW* 数字签名方案的一种攻击方法

注意到 $\boldsymbol{G},\boldsymbol{P},\boldsymbol{G}^*$ 为签名用户 A 的私钥。令 $\boldsymbol{A} = \boldsymbol{GP}$，$\boldsymbol{B} = \boldsymbol{P} + \boldsymbol{G}^*\boldsymbol{GP}$，则由式(11.3)可得

$$C = \{\boldsymbol{EB} + f(M,\boldsymbol{E})\boldsymbol{A} + \boldsymbol{ZW}^*\}\boldsymbol{W} + \boldsymbol{Z}$$

所以可以把$(\boldsymbol{A},\boldsymbol{B},\boldsymbol{W})$作为 AW 数字签名方案的签名私钥，而利用上式产生签名。容易验证它满足 $\boldsymbol{WW}^* = I_n$ 以及

$$A(\boldsymbol{H}',\boldsymbol{G}') = (0,I_k), \quad B(\boldsymbol{H}',\boldsymbol{G}') = (\boldsymbol{H}^{\mathrm{T}},0) \tag{11.5}$$

定理 11.5.1 如果一组矩阵$(\tilde{\boldsymbol{A}},\tilde{\boldsymbol{B}},\tilde{\boldsymbol{W}})$满足条件：

$$\tilde{\boldsymbol{W}}\boldsymbol{W}^* = I_n, \quad \tilde{\boldsymbol{A}}(\boldsymbol{H}',\boldsymbol{G}') = (0,I_k), \quad \tilde{\boldsymbol{B}}(\boldsymbol{H}',\boldsymbol{G}') = (\boldsymbol{H}^{\mathrm{T}},0) \tag{11.6}$$

则该组矩阵可作为 AW 数字签名方案的签名私钥成功地伪造签名。

证明：利用$(\tilde{\boldsymbol{A}},\tilde{\boldsymbol{B}},\tilde{\boldsymbol{W}})$作为私钥来构造签名。随机地选取一个 Hamming 重量 $W_{\mathrm{H}}(\boldsymbol{E}) \leqslant t$ 的向量 $\boldsymbol{E} \in F_2^n$ 和一个 Hamming 重量 $W_{\mathrm{H}}(Z) \approx l/2$ 的向量 $\boldsymbol{Z} \in F_2^l$。对任意 k 比特长的消息 M 的伪造签名是长度为 l 的二元序列：

$$\tilde{C} = \{\boldsymbol{E}\tilde{\boldsymbol{B}} + f(M,\boldsymbol{E})\tilde{\boldsymbol{A}} + \boldsymbol{ZW}^*\}\boldsymbol{W} + \boldsymbol{Z}$$

用户 B 接收到$(M,\tilde{C})$之后，实施以下步骤验证签名。首先计算

$$\begin{aligned} \tilde{\boldsymbol{V}} = \tilde{C}\boldsymbol{W}^* &= \{\boldsymbol{E}\tilde{\boldsymbol{B}} + f(M,\boldsymbol{E})\tilde{\boldsymbol{A}} + \boldsymbol{ZW}^*\}\tilde{\boldsymbol{W}}\boldsymbol{W}^* + \boldsymbol{ZW}^* \\ &= \boldsymbol{E}\tilde{\boldsymbol{B}} + f(M,\boldsymbol{E})\tilde{\boldsymbol{A}} \end{aligned}$$

然后计算伴随式，利用式(11.6)得

$$\tilde{\boldsymbol{V}}\boldsymbol{H}' = \boldsymbol{E}\tilde{\boldsymbol{B}}\boldsymbol{H}' + f(M,\boldsymbol{E})\tilde{\boldsymbol{A}}\boldsymbol{H}' = \boldsymbol{EH}^{\mathrm{T}}$$

从而由 Berlekamp-Massey 算法译码即可得到该随机向量 $\boldsymbol{E} \in F_2^n$。另一方面，再次利用式(11.6)得

$$\tilde{\boldsymbol{V}}\boldsymbol{G}' = \boldsymbol{E}\tilde{\boldsymbol{B}}\boldsymbol{G}' + f(M,\boldsymbol{E})\tilde{\boldsymbol{A}}\boldsymbol{G}' = f(M,\boldsymbol{E})$$

易见利用 M 和译码得到的 $\boldsymbol{E}$ 作为输入计算出的 $f(M,\boldsymbol{E})$ 即为上式的计算结果，故对消息 M 的伪造签名 $(M,\widetilde{\boldsymbol{C}})$ 有效。

等价私钥 $(\widetilde{\boldsymbol{A}},\widetilde{\boldsymbol{B}},\widetilde{\boldsymbol{W}})$ 的构造：

由于 $\widetilde{\boldsymbol{W}}$ 满足 $\widetilde{\boldsymbol{W}}\boldsymbol{W}^*=I_n$，故由公钥 $\boldsymbol{W}^*$ 解方程组即可得到 $\widetilde{\boldsymbol{W}}$（取任意一组解即可）。设 $\boldsymbol{K}=(\boldsymbol{H}',\boldsymbol{G}')$，则存在满秩矩阵 $\boldsymbol{R}$ 和 $\boldsymbol{Q}$ 使得 $\boldsymbol{RKQ}=\begin{pmatrix} I_r & 0 \\ 0 & 0 \end{pmatrix}$（其中 r 为矩阵 $\boldsymbol{K}$ 的秩）。取 $\boldsymbol{K}^*=\boldsymbol{Q}\begin{pmatrix} I_r & 0 \\ 0 & 0 \end{pmatrix}\boldsymbol{R}$ 为 $\boldsymbol{K}$ 的广义逆矩阵，即满足 $\boldsymbol{KK}^*\boldsymbol{K}=\boldsymbol{K}$。令

$$\widetilde{\boldsymbol{A}}=(0,I_k)\boldsymbol{K}^*,\quad \widetilde{\boldsymbol{B}}=(\boldsymbol{H}^{\mathrm{T}},0)\boldsymbol{K}^*$$

则利用式(11.5)可得

$$\begin{aligned}\widetilde{\boldsymbol{A}}(\boldsymbol{H}',\boldsymbol{G}') &= \widetilde{\boldsymbol{A}}\boldsymbol{K}=(0,I_k)\boldsymbol{K}^*\boldsymbol{K}=\boldsymbol{A}(\boldsymbol{H}',\boldsymbol{G}')\boldsymbol{K}^*\boldsymbol{K}\\ &= \boldsymbol{AKK}^*\boldsymbol{K}=\boldsymbol{AK}=(0,I_k)\\ \widetilde{\boldsymbol{B}}(\boldsymbol{H}',\boldsymbol{G}') &= \widetilde{\boldsymbol{B}}\boldsymbol{K}=(\boldsymbol{H}^{\mathrm{T}},0)\boldsymbol{K}^*\boldsymbol{K}=\boldsymbol{B}(\boldsymbol{H}',\boldsymbol{G}')\boldsymbol{K}^*\boldsymbol{K}\\ &= \boldsymbol{BKK}^*\boldsymbol{K}=\boldsymbol{BK}=(\boldsymbol{H}^{\mathrm{T}},0)\end{aligned}$$

所以 $(\widetilde{\boldsymbol{A}},\widetilde{\boldsymbol{B}},\widetilde{\boldsymbol{W}})$ 就是满足式(11.6)的一组矩阵，并且只利用签名用户的公钥信息 $\boldsymbol{G}'$，$\boldsymbol{H}'$，$\boldsymbol{W}^*$，$\boldsymbol{H}$ 就可以计算出 $(\widetilde{\boldsymbol{A}},\widetilde{\boldsymbol{B}},\widetilde{\boldsymbol{W}})$。

11.6　注记

本章重点介绍了一些在信息安全研究中常用的纠错码方法和技术，同时用典型实例阐述了纠错码方法和技术在信息安全领域中的应用。纠错码是一门发展相对比较成熟的学科，有着丰富的研究成果和广泛的应用。有很多纠错码方面的著作，如文献[1]～[8]，尤其是文献[1]和[2]是非常经典的著作，从事这一领域的研究工作者应该认真研读。关于纠错码方法和技术在信息安全领域中的应用文献很多，如文献[9]～[12]。本章的主要目的是为了满足纠错码方法和技术在信息安全领域中的应用而选材的，所以纠错码的一些新进展，如代数几何码、量子纠错码都没有涉及，感兴趣的读者可参阅文献[6]和[8]中的相关章节及其所引用的一些参考文献。纠错码方面的一些最新进展可在《IEEE on Information Theory》上找到，这也是纠错码研究领域的顶级刊物。

参 考 文 献

[1] MacWilliams F J, Sloane N J A. The Theory of Error-Correcting Codes, Amsterdam: North-Holland, 1977

[2] Berlekamp E R. Algebraic Coding Theory, New York: McGraw-Hill, New York, 1968

[3] Van Lint J H. Introduction to Coding Theory, Berlin: Springer-Verlag, 1982

[4] Susan Loepp, William K. Wootters. Protecting Information-From Classical Error Correction to Quantum Cryptography, United Kingdom: Cambridge, 2006

[5] 万哲先. 代数和编码. 北京：科学出版社，1985

[6] 王新梅，肖国镇. 纠错码——原理与方法. 西安：西安电子科技大学出版社，1991

[7] 肖国镇，卿斯汉. 编码理论. 北京：国防工业出版社，1993

[8] 冯克勤. 纠错码的代数理论. 北京：清华大学出版社，2005

[9] 王新梅，马文平，武传坤. 纠错密码理论. 北京：人民邮电出版社，2001

[10] McEliece R J. A public-key cryptosystem based on algebraic coding theory，DSN Progress Report 42-44，Jet Propulsion Laboratory，Pasadena，1978

[11] 冯登国，肖国镇. 对偶距离和相关免疫阶，通信学报，Vol. 15，No. 1，15-16，1994

[12] 张振峰，冯登国，戴宗铎. 基于纠错码的 AW 数字签名方案的分析，中国科学（E 辑），Vol. 32，No. 2，164-167，2003

第 12 章　图论方法与技术

图论是建立和处理各种数学模型的重要工具，在计算机科学、信息论、控制论、信息安全等领域有着广泛的应用。图论的研究源于著名的哥尼斯堡七桥问题，瑞士数学家 Euler 在 1736 年解决了这个问题，发表了图论的首篇论文“哥尼斯堡七桥问题无解”。图论诞生后并未及时获得足够的发展。直到 1936 年，匈牙利数学家 Konig 出版了图论的第一部专著《有限图与无限图理论》，这是图论发展史上的一个重要里程碑，它标志着图论进入突飞猛进发展的新阶段。本章的重点是介绍一些在信息安全研究中常用的图论方法和技术，包括图论的基本概念、一些重要的图、图的同构及典型应用实例。

12.1　基本概念

本节介绍图论的一些基本概念。

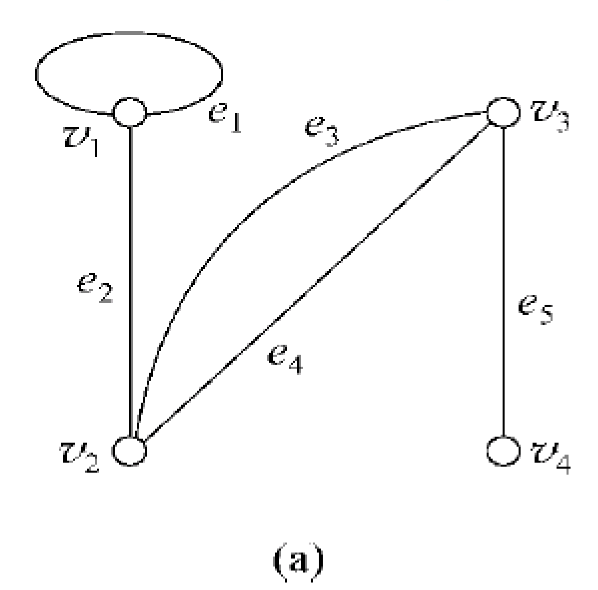
12.1.1　图的定义和示例

定义 12.1.1　一个图 G 定义为一个有序二元组 (V,E)，记为 $G=(V,E)$，其中：

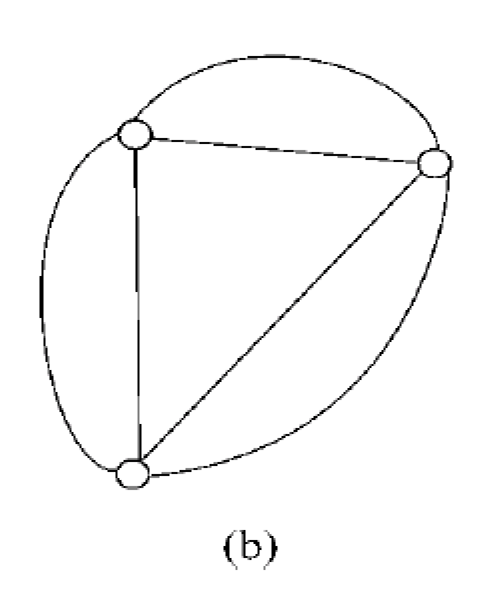
(1) $V=\{v_1,v_2,\cdots,v_n\}$ 是一个有限非空集合，V 称为 G 的**顶点集**，V 中元素称为 G 的**顶点**；

(2) E 是由 V 中的点组成的无序点对构成的集合，称为 G 的**边集**，E 中元素称为 G 的**边**，且同一点对在 E 中可出现多次。

习惯上用一图解来表示图，正是因为使用了这种图解式的表示法，图具有一种直观的外形。例如，$G_1=(V_1,E_1)$ 的图解如图 12.1(a) 所示，其中 $V_1=\{v_1,v_2,v_3,v_4\}$，$E_1=\{e_1=\{v_1,v_1\},e_2=\{v_1,v_2\},e_3=\{v_2,v_3\},e_4=\{v_2,v_3\},e_5=\{v_3,v_4\}\}$。如果两条边有相同的顶点，则称它们为**重边**，如图 12.1(a) 中的 e_3 和 e_4 所示。如果一条边的两个端点相同，则称为**环**，如图 12.1(a) 中的 e_1 所示。包含重边的图称为**多重图**，如图 12.1(b) 所示。不包含环和重边的图称为**简单图**，如图 12.1(c) 所示。



(a)


(b)

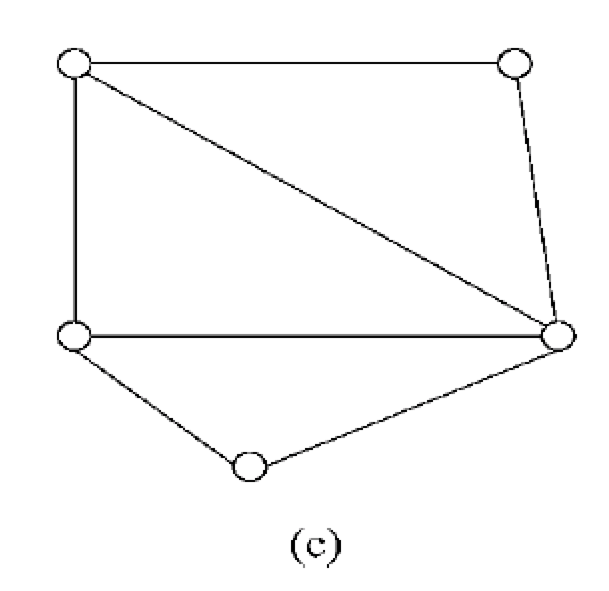

(c)

图 12.1　图解

设 $G=(V,E)$，V 是一个有限非空集合，E 是 V 中任意元素的有序二元对，那么称 G 为**有向图**。相对于有向图，前面的图称为**无向图**。类似地，可以定义**有向多重图**、**有向简单图**。一个含有 n 个顶点，m 条边的图也称为$(\boldsymbol{n},\boldsymbol{m})$**图**，其中$(n,0)$图称为**空图**，$(1,0)$图称为**平凡图**。

例 12.1.1 图 12.2 所示是一个$(4,6)$有向图；图 12.3(a)所示是一个$(5,7)$无向图；图 12.3(b)所示是一个空图；图 12.3(c)所示是一个平凡图。

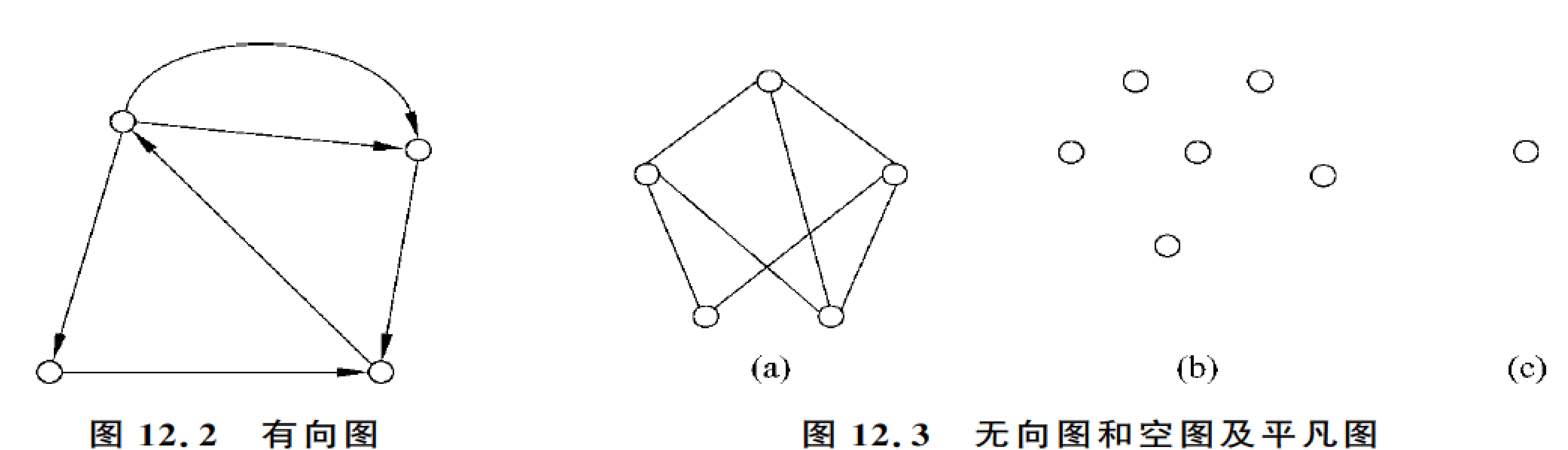


图 12.2 有向图　　图 12.3 无向图和空图及平凡图

一般情况下，若没有特殊声明，所谓的“图”均指**无向简单图**。

12.1.2 完全图和正则图

在图 G 中，顶点 u 和顶点 v 之间如果有一个边 $e=\{u,v\}$，则称 u 和 v 是**邻接顶点**，也称边 e **关联于** u 和 v。没有边关联的顶点称为**孤立点**。关联于公共顶点的相异边称为**邻接边**。

例 12.1.2 图 12.1(a) 中 v_1 和 v_2、v_2 和 v_3 分别是邻接顶点；边 $e_2=\{v_1,v_2\}$关联于点 v_1 和 v_2，边 $e_3=\{v_2,v_3\}$关联于点 v_2 和 v_3，因此 e_2 和 e_3 是邻接边。

定义 12.1.2 如果图 G 中任意两个不同顶点都是邻接顶点，则称图 G 是**完全图**。n 个顶点的完全图记为 K_n。

定义 12.1.3 由图 G 的所有顶点和那些为了使 G 成为完全图而需要添加的边组成的图称为 G 的**补图**，记为 $\overline{G}$。

例 12.1.3 图 12.4 中，图 12.4(a)所示是 5 个顶点的完全图 K_5，图 12.4(b)和(c)互为补图。

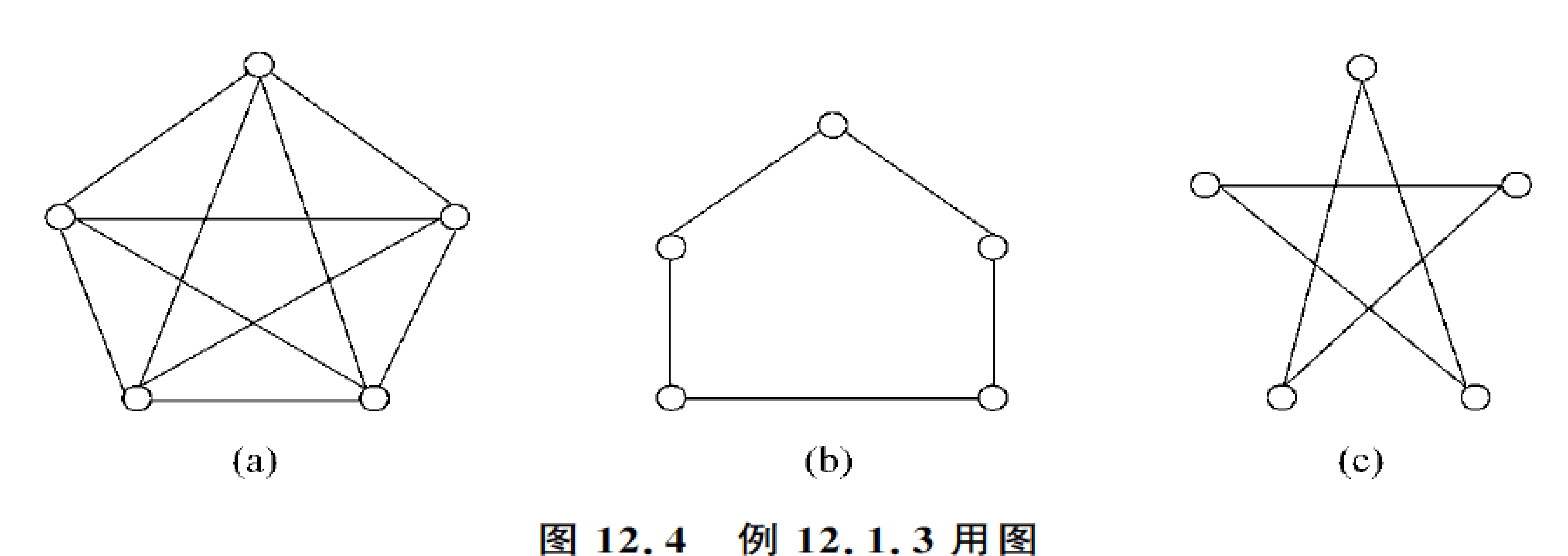


图 12.4 例 12.1.3 用图

定义 12.1.4 在图 G 中，与顶点 v 相关联的边的数目称为顶点 v 的**度**，记为 $d(v)$。在有向图中，顶点 v_i 和 v_j 分别称为边 $e=\{v_i,v_j\}$的**始点**和**终点**。以顶点 v 为始

点的边的数目称为 v 的**出度**，以顶点 v 为终点的边的数目称为 v 的**入度**，v 的入度和出度之和称为顶点 v 的度，记为 $d(v)$。

例 12.1.4　图 12.5(a)中，$d(v_1)=3$，$d(v_2)=3$，$d(v_3)=2$，$d(v_4)=2$，$d(v_5)=0$。图 12.5(b)中，$d(v_1)=5$，v_1 的入度为 2，出度为 3；$d(v_2)=4$，v_2 的入度为 2，出度为 2；$d(v_3)=3$，v_3 的入度为 0，出度为 3；$d(v_4)=4$，v_4 的入度为 3，出度为 1；$d(v_5)=2$，v_5 的入度为 2，出度为 0。

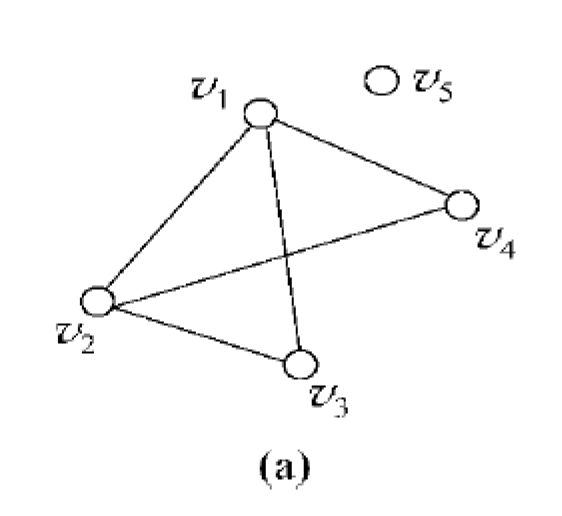
由于每条边关联于两个顶点，易见，一个图的所有顶点的度的总和等于边数的 2 倍，即有以下定理。

定理 12.1.1　设 $G=(V,E)$ 是一个 (n,m) 图，其中 $V=\{v_1,v_2,\cdots,v_n\}$，则

$$\sum_{i=1}^{n} d(v_i) = 2m$$

定义 12.1.5　一个无向图 G，如果它的所有顶点的度都为 k，则称图 G 是一个 **k 正则图**。

图 12.6 所示就是一个有 6 个顶点的 3 正则图。



(b)

图 12.5　例 12.1.4 用图

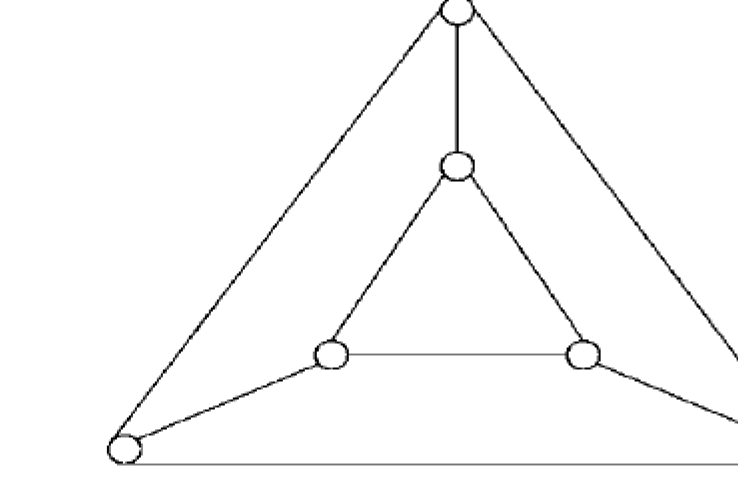


图 12.6　有 6 个顶点的 3 正则图

12.1.3　子图

正如集合有子集的概念一样，图也可相应地定义子图。

定义 12.1.6　设 $G=(V,E)$ 和 $\tilde{G}=(\tilde{V},\tilde{E})$ 是两个图，有：

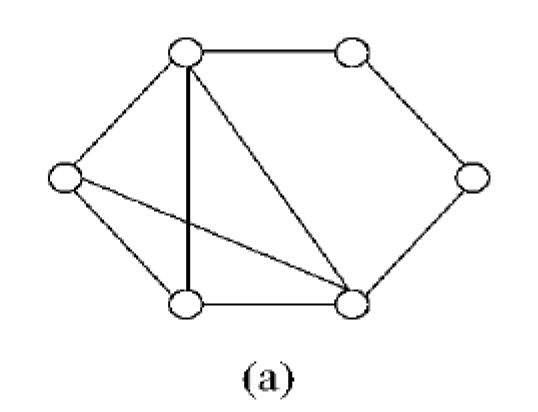
(1) 如果 $\tilde{V}\subseteq V$，$\tilde{E}\subseteq E$，则称 $\tilde{G}$ 是 G 的**子图**；

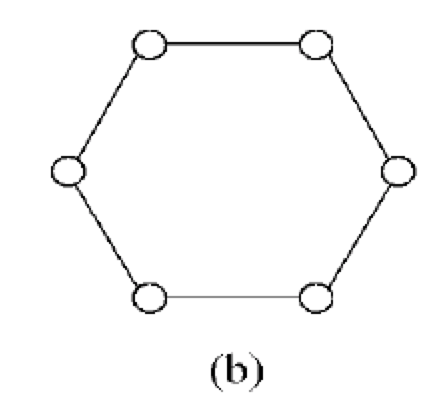
(2) 如果 $\tilde{V}\subseteq V$，$\tilde{E}\subset E$，则称 $\tilde{G}$ 是 G 的**真子图**；

(3) 如果 $\tilde{G}$ 是 G 的子图，且 $\tilde{V}=V$，则称 $\tilde{G}$ 是 G 的**生成子图**；

(4) 如果 $\tilde{G}$ 是 G 的子图，且 $\tilde{E}$ 由 E 中所有关联于 $\tilde{V}$ 中顶点的边组成，即 $\tilde{E}=\{e|e=(u,v)\in E,u,v\in\tilde{V}\}$，则称 $\tilde{G}$ 是 G 的由 $\tilde{V}$ 导出的子图，记为 $G[\tilde{V}]$，简称为 G 的**导出子图**。导出子图 $G[V\backslash\tilde{V}]$ 记为 $G-\tilde{V}$，它是 G 中删除 $\tilde{V}$ 中的顶点以及与这些顶点相关联的边所得到的子图。

显然，任一图 G 都是本身的子图。在图 12.7 中，图 12.7(b)、(c)都是图 12.7(a)的真子图，而且图 12.7(b)是图 12.7(a)的生成子图，图 12.7(c)是图 12.7(a)的导出子图。


(a)


(b)

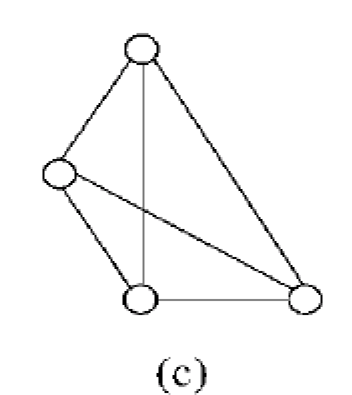

(c)

图 12.7　任一图 G 都是本身的子图

12.2　路与图的连通性

图的最基本性质是它的连通性。

定义 12.2.1　图 G 的一条**通路**是指一个有限非空序列 $w=v_0e_1v_1e_2v_2\cdots e_kv_k$，它的项交替地为顶点和边，使得对 $1\leqslant i\leqslant k$，e_i 关联于 v_{i-1} 和 v_i。顶点 v_0 和 v_k 分别称为 w 的**起点**和**终点**，整数 k 称为 w 的**长**。在简单图中，通路可简单地由其顶点序列来表示，即 $w=v_0v_1v_2\cdots v_k$。

由定义 12.2.1 可知，通路上的顶点和边均允许重复出现。在通路 $w=v_0e_1v_1e_2v_2\cdots e_kv_k$ 中，如果起点 v_0 与终点 v_k 相同，则称此通路为**回路**；如果起点 v_0 与终点 v_k 不同，则称此通路为**开路**。如果通路上的各边 $e_1,e_2,\cdots,e_k$ 互不相同，则称此通路为**简单通路**。如果通路上的顶点 $v_0,v_1,\cdots,v_k$ 互不相同，则称此通路为**基本通路**。如果回路上的各边互不相同，则称此回路为**简单回路**。如果回路上的顶点除 $v_0=v_k$ 之外互不相同，则称此回路为**基本回路**。

例 12.2.1　图 12.8 中，$v_1v_2v_4v_3v_1v_4$ 是从 v_1 到 v_4 的简单通路，但不是基本通路，它的长度为 5；$v_5v_6v_7v_5$ 既是简单回路又是基本回路，它的长度为 3；$v_2v_1v_4v_3v_5v_6v_4v_2$ 是简单回路，但不是基本回路，它的长度为 7。

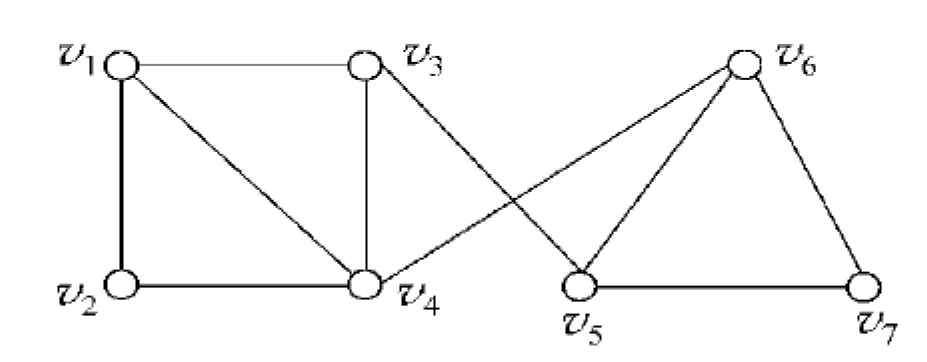


图 12.8　例 12.2.1 用图

定义 12.2.2　设 u 和 v 是图 G 的两个顶点，从 u 到 v 的最短通路的长度称为 u 和 v 之间的**距离**，用 $d(u,v)$ 表示；如果 u 和 v 是不连通的，则记 $d(u,v)=\infty$。

根据上面的定义，对图 G 中的任意连通的 3 个顶点 u、v、w，它们之间的距离满足下面的三角不等式

$$d(u,v)+d(v,w)\geqslant d(u,w)$$

定理 12.2.1　设图 G 的顶点集为 $V=\{v_1,v_2,\cdots,v_n\}$，则对图中任意连通的两个顶点 v_i 和 $v_j(v_i\neq v_j)$，它们之间的距离一定不大于 $n-1$。

证明：设 R 为从 v_i 到 v_j 的一条路，且

$$R=v_iv_{i_1}v_{i_2}\cdots v_{i_l}v_j$$

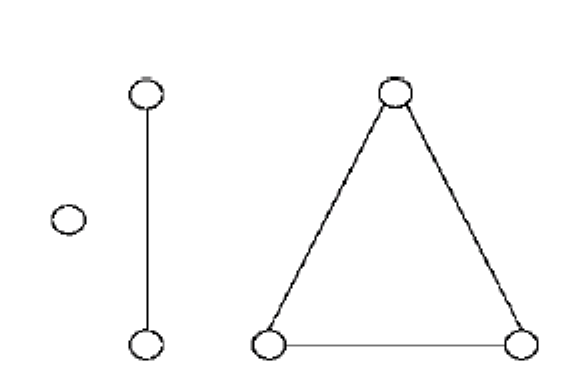
若 R 中存在两个相同的顶点 $v_{i_p}=v_{i_q}$，则 $v_{i_{p+1}}\cdots v_{i_q}$ 可以从 R 中删去，得到一条较短的路 $R'=v_iv_{i_1}v_{i_2}\cdots v_{i_p}v_{i_{q+1}}\cdots v_{i_l}v_j$。如果在 R' 中还存在相同的顶点，则可以重复上

面的操作来得到更短的路。容易看出,从 v_i 到 v_j 的最短路中肯定不会存在相同的顶点。设 v_i 到 v_j 的距离为 k,则最短路中顶点个数为 $k+1$,因此有 $k+1\leqslant n$,即 $k\leqslant n-1$,定理得证。

定义 12.2.3 如果图 $G=(V,E)$中存在着从顶点 v_i 到顶点 v_j 的通路,则称 v_i 和 v_j 是**连通的**。如果图 G 中任意两个顶点都是连通的,则称图 G 为**连通图**;否则称为**非连通图**。

由定义 12.2.3 可知,连通是顶点集 V 上的一个等价关系(满足自反性、对称性和传递性),于是可将 V 划分成等价类 $V_1,V_2,\cdots,V_k$。这样,两个顶点 u 和 v 是连通的当且仅当它们属于同一子集 V_i。子图 $G[V_1],G[V_2],\cdots,G[V_k]$称为 G 的**连通分支**,简称**分支**,其个数常记为 $w(G)$,即 $w(G)=k$。显然,若 G 是连通的,则 $w(G)=1$。

例 12.2.2 图 12.9 所示是一个非连通图,而图 12.10 所示是一个连通图。


图 12.9 非连通图

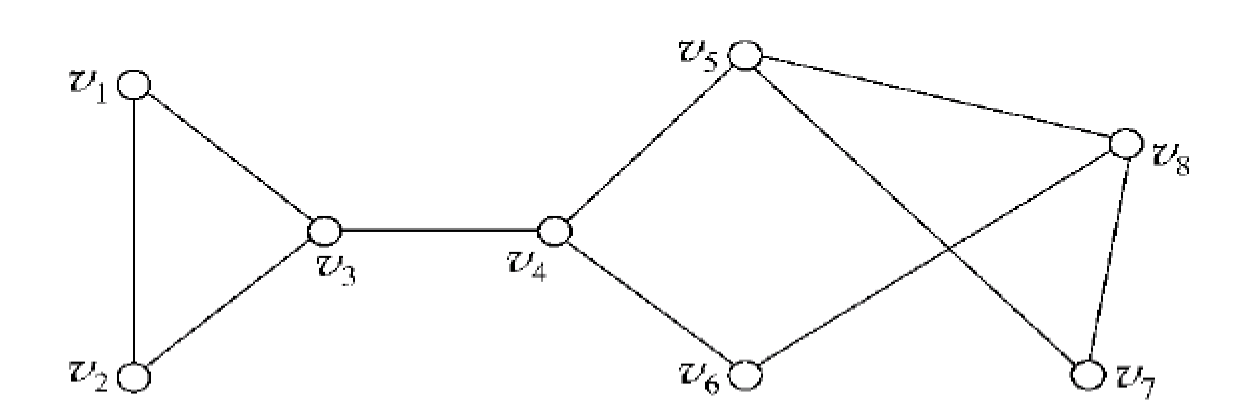


图 12.10 连通图

定义 12.2.4 若在图 G 中去掉一条边 e 后,图的分支数增加,则称此边 e 为图 G 的**割边**。

定义 12.2.5 若在图 G 中去掉一个顶点 v 和与 v 关联的所有边后,图的分支数增加,则称顶点 v 为图 G 的**割点**。

例 12.2.3 图 12.10 中,边$\{v_3,v_4\}$是割边,顶点 v_3、v_4 都是割点。

12.3 图的矩阵表示

一个图的矩阵表示不仅仅是给出图的一种表示方法,重要的是通过对矩阵的讨论,可以得到有关图的很多性质。此外,在图论的应用中,图的矩阵表示也具有重要的作用,可以提高计算机对图的处理能力。

定义 12.3.1 设 $G=(V,E)$是一个图,其中 $V=\{v_1,v_2,\cdots,v_n\}$,令

$$a_{ij}=\begin{cases}1 & \{v_i,v_j\}\in E\\ 0 & \{v_i,v_j\}\notin E\end{cases}$$

则 n 阶矩阵 $\boldsymbol{A}=(a_{ij})$称为图 G 的**邻接矩阵**。

定义 12.3.2 设 $G=(V,E)$是一个图,其中 $V=\{v_1,v_2,\cdots,v_n\}$,$E=\{e_1,e_2,\cdots,e_m\}$,令

$$b_{ij}=\begin{cases}1 & v_i\text{ 和 }e_j\text{ 关联}\\ 0 & v_i\text{ 和 }e_j\text{ 不相关联}\end{cases}$$

则矩阵 $\boldsymbol{B}=(b_{ij})_{n\times m}$称为图 G 的**关联矩阵**。

例 12.3.1 图 12.11 所示对应的邻接矩阵和关联矩阵分别为

$$
\boldsymbol{A}=\begin{array}{c}\\ v_1\\ v_2\\ v_3\\ v_4\end{array}\begin{array}{c}\begin{array}{cccc}v_1 & v_2 & v_3 & v_4\end{array}\\ \begin{bmatrix}0 & 1 & 1 & 1\\ 1 & 0 & 1 & 0\\ 1 & 1 & 0 & 1\\ 1 & 0 & 1 & 0\end{bmatrix}\end{array}\qquad
\boldsymbol{B}=\begin{array}{c}\\ v_1\\ v_2\\ v_3\\ v_4\end{array}\begin{array}{c}\begin{array}{ccccc}e_1 & e_2 & e_3 & e_4 & e_5\end{array}\\ \begin{bmatrix}1 & 1 & 0 & 0 & 1\\ 0 & 1 & 1 & 0 & 0\\ 0 & 0 & 1 & 1 & 1\\ 1 & 0 & 0 & 1 & 0\end{bmatrix}\end{array}
$$

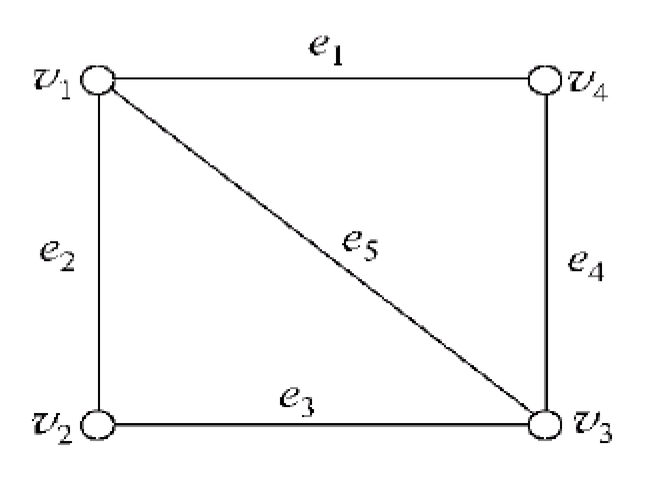


图 12.11 例 12.3.1 用图

显然,无向图的邻接矩阵和关联矩阵有以下性质:邻接矩阵一定是对称矩阵;对简单图来说,其邻接矩阵的主对角线元素必然全为 0;顶点 v_i 的度是邻接矩阵(关联矩阵)第 i 行元素中 1 的个数;关联矩阵每列中 1 的个数恰为 2,因为与每条边关联的顶点数恰为两个。一般来说,图的邻接矩阵比关联矩阵小得多。

定理 12.3.1 设图 $G=(V,E)$,$V=\{v_1,v_2,\cdots,v_n\}$,$\boldsymbol{A}=(a_{ij})$是 G 的邻接矩阵,则 $\boldsymbol{A}^l(l=1,2,\cdots)$中第 i 行第 j 列元素 a_{ij}^l 等于 v_i 和 v_j 之间长度为 l 的通路的数目。

证明: 设 $\boldsymbol{A}^l=(a_{ij}^l)$,用归纳法证明。

当 $l=1$ 时,根据邻接矩阵的定义,结论成立。

假设当 $l=k$ 时,结论成立,即 v_i 和 v_j 之间长度为 k 的通路的数目为 a_{ij}^k。由矩阵乘法的定义,矩阵 $\boldsymbol{A}^{k+1}$ 中的元素 a_{ij}^{k+1} 是 $\boldsymbol{A}^k$ 的第 i 行向量和 $\boldsymbol{A}$ 的第 j 列向量的点积,即 $a_{ij}^{k+1}=\sum_{s=1}^{n}a_{is}^k a_{sj}$,其中 a_{is}^k 是从 v_i 和 v_s 之间的长为 k 的路的数目。如果 $a_{sj}=1$,则 $a_{is}^k a_{sj}$ 表示从 v_i 和 v_j 之间的长度为 $k+1$,且 v_j 前一个顶点为 v_s 的路的数目。当 s 从 1 遍历到 n 时,考虑了所有 v_i 和 v_j 之间的长度为 $k+1$ 的路,因此当 $l=k+1$ 时结论也成立。

综上所述,结论成立,证毕。

邻接矩阵可用于判断图的连通性。进一步,有以下定理。

定理 12.3.2 设图 $G=(V,E)$,$V=\{v_1,v_2,\cdots,v_n\}$,$\boldsymbol{A}$ 是 G 的邻接矩阵,则 G 是连通的当且仅当$(\boldsymbol{I}+\boldsymbol{A})^{n-1}>0$,其中 $\boldsymbol{I}$ 是 n 阶单位矩阵。

证明: 假设$(\boldsymbol{I}+\boldsymbol{A})^{n-1}>0$,由于

$$(\boldsymbol{I}+\boldsymbol{A})^{n-1}=I+C_{n-1}^1\boldsymbol{A}+C_{n-1}^2\boldsymbol{A}^2+\cdots+\boldsymbol{A}^{n-1}$$

所以对任何 i 和 j,$i\neq j$,$1\leqslant i,j\leqslant n$,存在 $1\leqslant l\leqslant n-1$,使 $a_{ij}^l>0$,这里 a_{ij}^l 是 $\boldsymbol{A}^l$ 中第 i 行第 j 列元素。即 v_i 和 v_j 之间存在通路,所以 G 是连通的。

反之,假设 G 是连通的,则任意两个不同的顶点 v_i 和 v_j 之间必存在通路。于是,必存在 $1\leqslant l\leqslant n-1$,满足 $a_{ij}^l>0$,从而 $\sum_{l=0}^{n-1}\boldsymbol{A}^l$ 中第 i 行第 j 列元素大于 0。故$(\boldsymbol{I}+\boldsymbol{A})^{n-1}>0$。定理得证。

12.4 Euler 图与 Hamilton 图

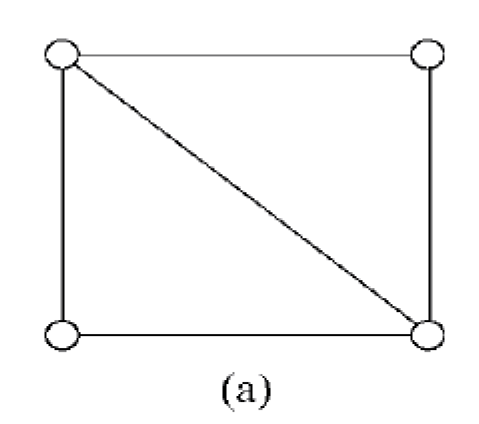
本节讨论两种典型的图——Euler 图与 Hamilton 图。

1736 年,瑞士数学家 Euler 在解决哥尼斯堡七桥问题时形成了 Euler 图的概念。

下面给出有关定义及定理。

定义 12.4.1　无向图 G 中的回路 α，若 G 中的每条边都在 α 中出现一次且仅一次，则称 α 为 **Euler 回路**。具有 Euler 回路的图称为 **Euler 图**。

例 12.4.1　图 12.12(a) 所示不是一个 Euler 图，而图 12.12(b)所示则是一个 Euler 图。显然，Euler 图肯定是连通图。如何判断一个连通图是否为 Euler 图，有以下定理。


(a)

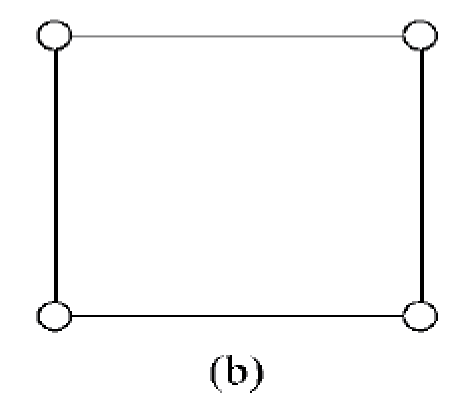

(b)

图 12.12　Euler 图与非 Euler 图

定理 12.4.1　无向连通图 $G=(V,E)$是 Euler 图当且仅当 G 的所有顶点的度都是偶数。

证明：若连通图 G 是一个 Euler 图，设它的一条 Euler 回路为 α。因为 G 的每条边在 α 中出现且仅出现一次，故 α 必通过 G 的每个顶点。当 α 通过某顶点时，进去一次出来一次，此顶点的度就增加 2，从而每个顶点的度都是偶数。

反之，假设连通图 G 的顶点的度数都是偶数，用归纳法证明 G 是 Euler 图。

当边数$|E|=0$ 时，由于 G 连通，G 只包含一个顶点，Euler 回路由此顶点构成。

假设当$|E|=k$ 时，结论成立。考虑$|E|=k+1$ 的情况，从图 G 中的任意一点 a 出发，经过每条边最多一次，沿着一条路一直走下去，对路上的顶点 $v\neq a$，由于 v 的度为偶数，因此总是可以从 v 中出来，直到最后回到 a，记这边回路为 C。如果 C 经过了图 G 的所有的顶点，那么它就是图 G 的一条 Euler 回路。否则，从图 G 中去掉 C 经过的边，得到一个新的图 $G'=H_1,H_2,\cdots,H_k$，其中 $H_1,H_2,\cdots,H_k$ 为图 G'的分支。因为图 G'的顶点度数仍为偶数，所以这些分支的顶点度数也为偶数，但它们的边数小于 $k+1$，因此它们都存在各自的 Euler 回路，分别记为 $C_1,C_2,\cdots,C_k$。图 G 为一连通图，所以 $C_1,C_2,\cdots,C_k$ 分别与 C 至少存在着一个公共顶点，通过这些顶点可以将 $C_1,C_2,\cdots,C_k$ 都加到 C 上，形成一条新的 Euler 回路，它经过了图 G 的所有边。即当$|E|=k+1$ 时，结论也成立，证毕。

例 12.4.2　哥尼斯堡七桥问题如图 12.13 所示。普雷格尔河流经哥尼斯堡城，把哥尼斯堡城分成 4 个城区，人们在河上架设 7 座桥以方便市民在城区之间穿行。图 12.13(a)给出了哥尼斯堡城的一个地图，图中标出了 4 个城区(A,B,C,D)、河以及 7 座桥的位置。哥尼斯堡七桥问题是：能否从一点出发，走遍 7 座桥，且通过每座桥恰好一次，最后仍回到起始地点。该问题可用图 12.13(b)所示来表述。根据定理 12.4.1，由于图中所有顶点度数均为奇数，故不是 Euler 图，从而哥尼斯堡七桥问题无解。

定义 12.4.2　对于一个连通图 G，含有 G 的每条边恰一次的开路称为图 G 的 **Euler 路**。

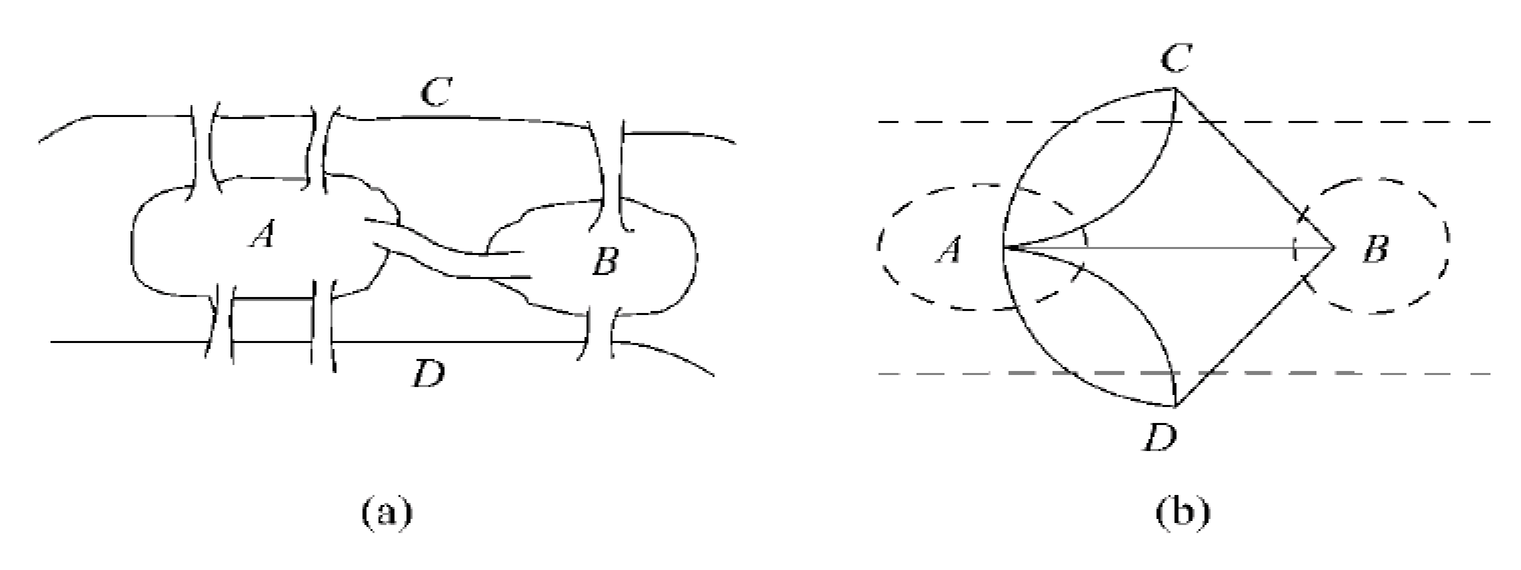


图 12.13　哥尼斯堡七桥问题

在定理 12.4.1 的帮助下，可以轻松地刻画含有 Euler 路的图。

推论 12.4.1　一个连通图 G 含有一条 Euler 路当且仅当 G 恰有两个度为奇数的顶点。而且，G 的每一条 Euler 路始于一个度为奇数的顶点而终止于另一个度为奇数的顶点。

证明：首先讨论必要性。假设 G 含有一条 Euler 路 T。因此 T 是某两个互异顶点 v_i 和 v_j 之间的一条通路，即 $T=v_i,\cdots,v_j$。现在由 G 通过添加一个新的度为 2 的顶点 u 且分别连接 u 到 v_i 和 v_j，从而构造成一个新的连通图 G'。则 $T'=v_i,\cdots,v_j,u,v_i$ 是 G' 的一条 Euler 回路。根据定理 12.4.1，G' 中每个顶点的度均为偶数，所以在 G 中只有 v_i 和 v_j 的度为奇数。

用类似的方法讨论充分性。假设连通图 G 恰好有两个度为奇数的顶点 v_i 和 v_j。添加一个新的度为 2 的顶点 u 到 G 上，并分别连接 u 到 v_i 和 v_j，记所得到的新图为 G'。因此 G' 是一个每个顶点度均为偶数的连通图。根据定理 12.4.1，G' 是一个含有 Euler 回路 C 的 Euler 图，则 C 一定经过 v_i 和 v_j。由于 u 的邻接顶点只有 v_i 和 v_j，则从 C 中删除顶点 u，得到一条从 v_i（或 v_j）出发而终止于 v_j（或 v_i）的 Euler 路。证毕。

Euler 图研究的是边的遍历问题，下面就来讨论顶点的遍历问题。

定义 12.4.3　无向图 G 中经过所有顶点一次的回路称为 **Hamilton 回路**，含有 Hamilton 回路的图称为 **Hamilton 图**。类似地，经过所有顶点恰一次的开路称为 **Hamilton 路**。

Hamilton 图的概念是爱尔兰数学家 Hamilton 于 1859 年引入的，他用正十二面体的 20 个顶点代表 20 个城市，要求从一个城市出发，经过每个城市恰好一次，然后回到出发城市。图 12.14 所示是一个正十二面体的展开图，按照图中的顶点编号所构成的回路，就是 Hamilton 回路的一个解。

与 Euler 图不同，到目前为止尚没有找到判别一个图是否是 Hamilton 图的有效充要条件。这是图论中未解决的重要难题之一。

定理 12.4.2　若图 $G=(V,E)$ 为 Hamilton 图，则对 V 的每个非空子集 S 均有

$$w(G-S)\leqslant|S|$$

证明：设 $C=v_0v_1v_2\cdots v_nv_0$ 为图 G 的 Hamilton 回路，若删去 v_i，则 $C-\{v_i\}$ 为一条开路，所以 $w(C-\{v_i\})=1$。易见，对于 $C-S$，随着 S 增加一个顶点，$w(C-S)$ 最多增加 1，通过归纳可知，$w(C-S)\leqslant|S|$。

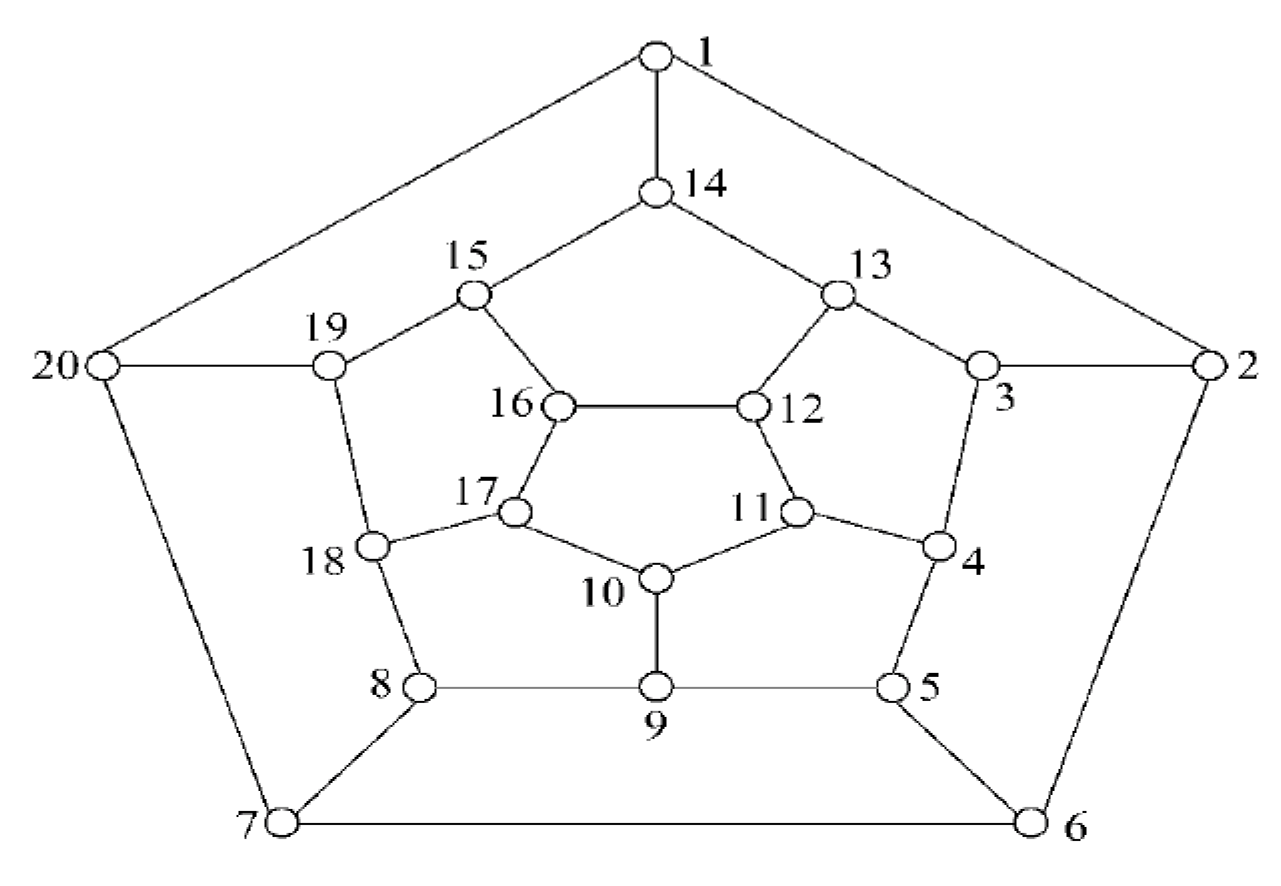


图 12.14　Hamilton 图

另外，$C-S$ 为 $G-S$ 的生成子图，因此 $w(G-S)\leqslant w(C-S)$，所以

$$w(G-S)\leqslant |S|$$

证毕。

定理 12.4.2 是判断 Hamilton 图的必要条件，可以用来排除一些图为 Hamilton 图的可能。根据定理，如果 $w(G-S)\geqslant|S|$，那么图 G 肯定不是 Hamilton 图。

例 12.4.3　图 12.15 中，左边的图满足 $w(G-S)\geqslant|S|$，它不是 Hamilton 图；右边的图满足 $w(G-S)\leqslant|S|$，但它也不是 Hamilton 图。

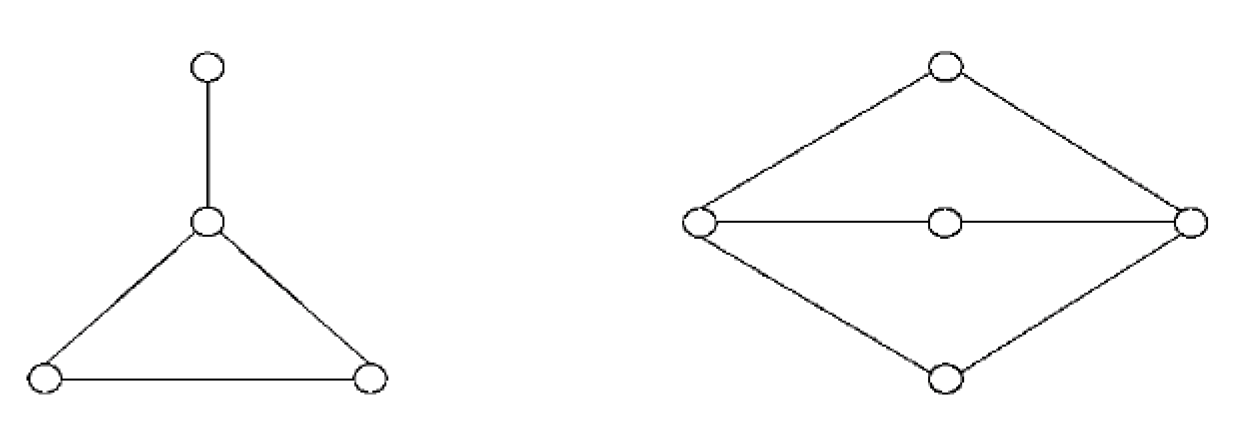


图 12.15　非 Hamilton 图

定理 12.4.3　(n,m) 图 $G=(V,E)$ 为 $n\geqslant3$ 的图，如果对于 G 的每对不邻接的顶点 u,v 都有 $d(v)+d(u)\geqslant n$，则 G 为 Hamilton 图。

证明：用反证法证明。假设存在一个 $n\geqslant3$ 的非 Hamilton 图 G，使得对于 G 的每对不邻接的顶点 u,v，均有 $d(v)+d(u)\geqslant n$。对图 G 做以下处理，在保证所得的新图是非 Hamilton 图的前提下，向 G 中添加边直到不能再添加为止，得到的新图记为 H。对于 H 中每对不邻接的顶点 u,v，一定有 $d(v)+d(u)\geqslant n$。

因为 H 不是完全图，所以 H 含有不邻接的顶点对。设 x 和 y 是 H 的不邻接的顶点对。显然，图 $H+\{x,y\}$ 是一个 Hamilton 图，且 $H+\{x,y\}$ 的每个 Hamilton 回路必包含边 $\{x,y\}$。也就是说，在 H 中存在一个从 x 到 y 的 Hamilton 路，记为 $x=v_1,v_2,\cdots,v_n=y$。可以得到：如果 xv_i 是 H 的一条边，其中 $2\leqslant i\leqslant n$，则 $v_{i-1}y$ 不是 H 的一条边；否则，$x,v_i,v_{i+1},\cdots,y,v_{i-1},v_{i-2},\cdots,x$ 是 H 的 Hamilton 回路，这是不可能的。因此，对于 $\{v_2,v_3,\cdots,v_n\}$ 中的每个与 x 邻接的顶点，在 $\{v_1,v_2,\cdots,v_{n-1}\}$ 中都有一个顶点与 y 不邻接。即 $d(y)\leqslant(n-1)-d(x)$，故

$$d(y)+d(x)\leqslant n-1$$

与假设矛盾。定理得证。

定理 12.4.4 设 u 和 v 是图 $G=(V,E)$ 的两个不邻接的顶点,并且满足 $d(u)+d(v)\geqslant|V|$,则图 G 是 Hamilton 图的充要条件是 $G+\{u,v\}$ 是 Hamilton 图。

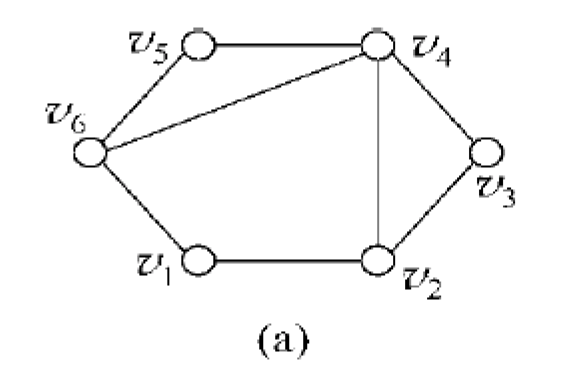
证明:必要性是显然的。下面证明充分性。

反证法。假设 $G+\{u,v\}$ 是 Hamilton 图,u 和 v 是图 G 的两个不邻接的顶点,但 G 不是 Hamilton 图。那么推出 $G+\{u,v\}$ 的每个 Hamilton 回路都必须含有边 $\{u,v\}$。因此,G 含有一条 u 和 v 之间的 Hamilton 路。因为 $d(u)+d(v)\geqslant|V|$,由定理 12.4.3 的证明知,G 含有一个 Hamilton 回路,导致矛盾。

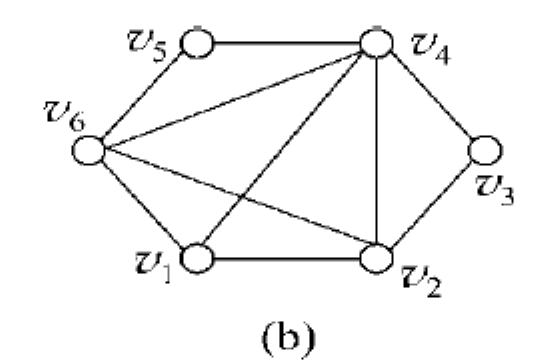
进一步可以引入图的闭包的定义。

定义 12.4.4 图 $G=(V,E)$ 的**闭包** G_c 是指按以下操作所获得的图:由 G 出发递归地连接度和至少为 $|V|$ 的不邻接顶点对,每一步都是针对前一步所获得的图,直到没有这样的顶点对为止。

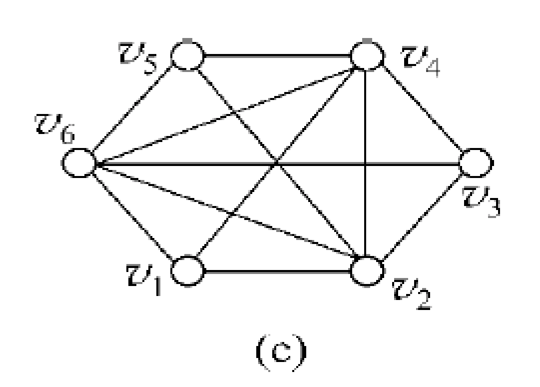
例 12.4.4 图 12.16 所示说明了图 G 的闭包的构造过程。



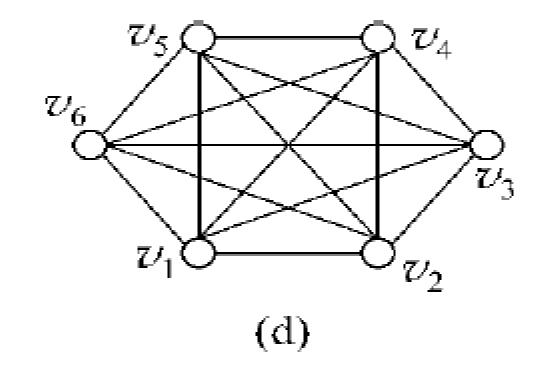
(a)



(b)



(c)



(d)

图 12.16 例 12.4.4 用图

重复应用定理 12.4.4,可以得到下面结果。

定理 12.4.5 图 G 是 Hamilton 图当且仅当它的闭包 G_c 是 Hamilton 图。

12.5 树

树是图论中的一种简单而重要的图,它的应用非常广泛。本节将讨论树的一些基本性质。

定义 12.5.1 不含回路的无向连通图称为**无向树**,简称**树**,常用 T 表示。连通分支数大于 1,且每个连通分支都是树的无向图称为**森林**。树中的边称为**树枝**,树中度为 1 的顶点称为**树叶**。

例 12.5.1 如图 12.17 所示,3 棵包含 4 个顶点的树。这些树的任意一种组合均为一个森林。

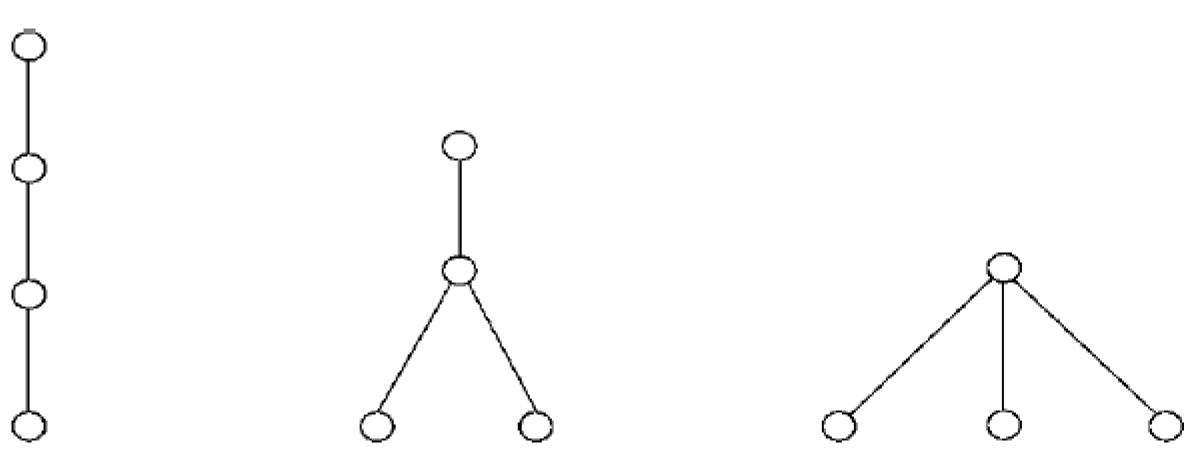


图 12.17 例 12.5.1 用图

树有许多等价的定义，在下面的定理中列出了其中的 5 个等价命题。

定理 12.5.1　设无向图 $G=(V,E)$是一个(n,m)图，则下列命题等价：

(1) G 是树；

(2) G 中任意两顶点间有且仅有一条路相连；

(3) G 是连通的，且 $m=n-1$；

(4) G 无回路，且 $m=n-1$；

(5) G 无回路，但在 G 中任意不相邻两顶点间增加一条边，就得到唯一一个回路。

证明：(1)⇒(2)：反证法证明。图 G 为树，由于连通性，任意两顶点之间必有路相连。假设 u、v 间存在着两条路 α 和 β，其中 $\alpha=ua_1a_2\cdots a_sv$，$\beta=ub_1b_2\cdots b_tv$。令 k 是使 $a_{k+1}\neq b_{k+1}$成立的最小整数。由于 α、β 最后的顶点都是 v，因此一定存在两个 i，$j>k$ 使得 $a_i=b_j$，那么 $a_ka_{k+1}\cdots a_ib_{j-1}\cdots a_k$ 为图 G 的一个回路，与 G 为树相矛盾。

(2)⇒(3)：由于图 G 中任意两顶点之间有路相连，所以 G 是连通的。对顶点数 n 进行归纳，证明 $m=n-1$。

当 $n=1,2$ 时，结论显然成立。

假设当 $n\leqslant k(k\geqslant 2)$时，结论都成立。当 $n=k+1$ 时，从 G 中随便去掉一条边，由(2)，去掉该边后，G 得到两个连通分支 G_1 和 G_2。设 G_1 和 G_2 的顶点数分别为 n_1 和 n_2，边数分别为 m_1 和 m_2。由归纳假设：

$$m_1=n_1-1,\quad m_2=n_2-1$$

那么 $m_1+m_2=n_1+n_2-2$，又 $m=m_1+m_2+1$，$n=n_1+n_2$，所以 $m=n-1$。

(3)⇒(4)：只须证明 G 无回路。反证法证明。假设 G 中存在一长度为 k 的回路 C，则 C 上 k 个顶点和 k 条边。对于 $n-k$ 个不在 C 上的每个顶点 v_i，必有一条关联于它的边 e_i，且 e_i 在连接 v_i 与 C 上顶点的最短通路上，这种边每条都不相同，因此 $m\geqslant n$有去掉回路中的一条边，使图的顶点数与连通性不变。导出矛盾。

(4)⇒(5)：先证明 G 是连通的。设 G 有 k 个分支，由(4)，G 中无回路，故每个分支是一棵树。设这 k 棵树分别是 $G_1,G_2,\cdots,G_k$，且 G_i 有 n_i 个顶点，m_i 条边，$1\leqslant i\leqslant k$，从而 $m_i=n_i-1$，于是，$m=n_1+n_2+\cdots+n_k-k=n-k$。由(4)得出 $k=1$。因此 G 是连通的，于是 G 是连通而无回路的图，(1)成立，从而(2)成立。

设 u 和 v 是 G 中不邻接的两个顶点，由(2)，u 和 v 之间存在一条通路 P，路 P 和边$\{u,v\}$构成图 $G+\{u,v\}$的一个回路。因为 G 中无回路，所以 $G+\{u,v\}$的任何一个回路含有边$\{u,v\}$。于是 $G+\{u,v\}$中若有两个不同的回路，可得到 G 中 u 和 v 之间的两条不同的通路，与(2)矛盾。

(5)⇒(1)：若 G 不连通，则存在顶点 u 和 v，使得 $G+\{u,v\}$中不含回路，这与(5)矛盾。因此 G 是连通的无回路的图，从而(1)成立。

定理 12.5.2　具有两个及以上顶点的树至少有两片树叶。

证明：假设(n,m)图 G 为树，且 $n\geqslant 2$，G 的所有顶点的度数之和为 S，则有 $S=2m$，又 $m=n-1$，所以 $S=2n-2$。

如果 G 中只有一顶点为树叶，即其他 $n-1$ 个顶点的度数都不小于 2，$S\geqslant$

$2(n-1)+1>2n-2$，与 $S=2n-2$ 矛盾。因此 G 中至少有两片树叶。证毕。

定义 12.5.2 若 T_G 是无向图 G 的生成子图而且又是树，则称 T_G 是图 G 的**生成树**。

例 12.5.2 在图 12.18 中，图 T_1 和图 T_2 是图 G 的两个不同生成树。

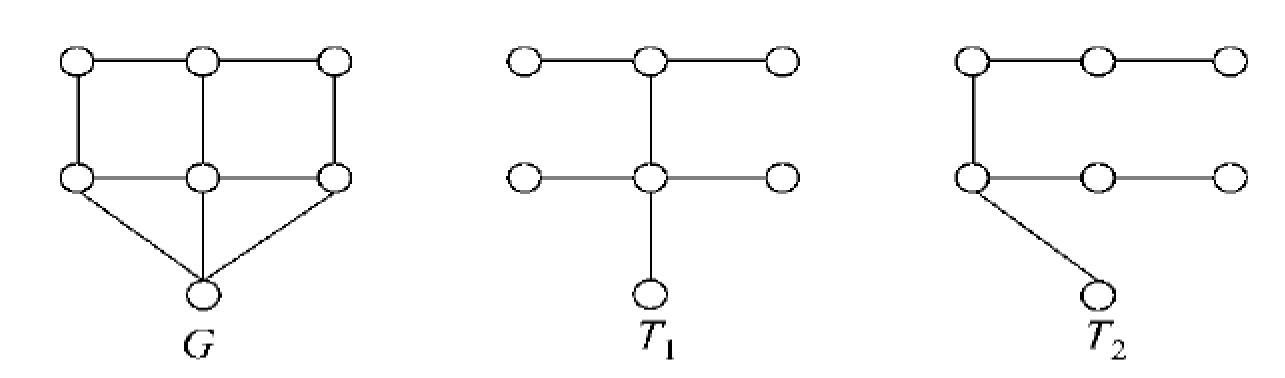


图 12.18 例 12.5.2 用图

定理 12.5.3 每个连通图 G 至少包含一棵生成树。

构造生成树的最简单方法是破圈法。

破圈法 设 G 是一连通图，在图 G 中任取一个回路，去掉该回路上一条边，所得图仍为连通图，这样进行下去，最终可得到一个无回路连通生成子图，即得到 G 的一棵生成树。

破圈法实际上已给出定理 12.5.3 的一个证明。不难发现，一个连通图的生成树一般不是唯一的，除非 G 本身是树。

定义 12.5.3 设 G 是一连通图，对它的每条边 e 都分配一个数值，该数值称为边的**权值**，记为 $w(e)$。图 G 称为**赋权图**。

定义 12.5.4 设 G 是一赋权图，G 的生成树所有树枝上权的总和，称为生成树的权。权值最小的生成树称为 G 的**最小生成树**。

许多实际问题常常化为求赋权图中的最小生成树问题。最小生成树问题的求解方法很多，其中最为著名的一个算法是由 Kruskal 提出来的。

Kruskal 算法 对于一个(n,m)连通赋权图 G，G 的最小生成树 T_G 按下述方法构造：首先将 G 中不是自回路的边按权值递增的顺序排列成 $L=a_1,a_2,\cdots,a_t(t\leqslant m)$。取 $e_1=a_1$；在 L 中剩下的边中按顺序选择边，使其不与前面所选的边构成回路，作为 T 的第二条边 e_2。这样的过程继续下去，直至找出边 $e_1,e_2,\cdots,e_{n-1}$ 构成 G 的一棵生成树 T_G。

定理 12.5.4 对于一个(n,m)连通赋权图 G，Kruskal 算法构成的生成树是 G 的最小生成树。

证明：反证法证明。假设 T' 是 G 的最小生成树。树 T' 的边集 $E'=\{e'_1,e'_2,\cdots,e'_r\}$，假设 k 是使得 $f(e_k)\neq f(e'_k)$ 的最小整数，也就是说 $f(e_1)=f(e'_1),\cdots,f(e_{k-1})=f(e'_{k-1})$。

将 e_k 添加到 T' 的边集中，则 T' 一定会出现一个包含 e_k 的圈。由于 T 为树，因此这个圈中存在一条边 $e'_i\notin E_r$。然后从 T' 中去掉边 e'_i 得到一棵新的树 $T''=T'+e_k-e'_i$。又 T' 为最小生成树，所以 $f(T'')\geqslant f(T')$，即

$$f(e_k)-f(e'_i)\geqslant 0$$

但是根据 e_k 的选取原则有 $f(e_k)-f(e'_i)\leqslant 0$，所以有 $f(e_k)=f(e'_i)$。因此 T 也是最小生成树。

例 12.5.3　图 12.19 演示了如何应用 Kruskal 算法构造连通赋权图的一个最小生成树。

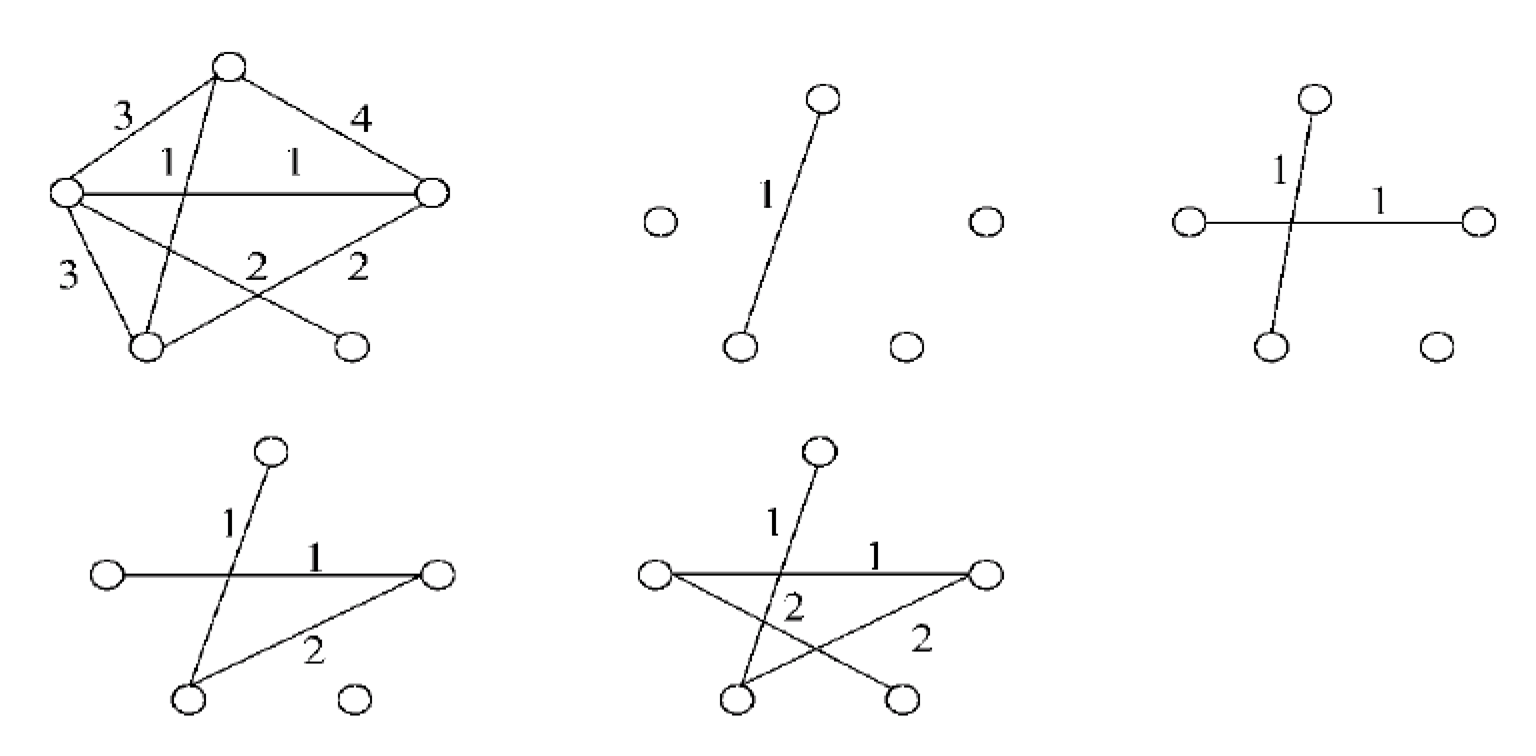


图 12.19　用 Kruskal 算法构造最小生成树

无向树的很多性质都可以推广到有向树中。

定义 12.5.5　一个有向图 G 在不考虑弧的方向时如果是一棵树，则称 G 为**有向树**。

定义 12.5.6　如果有向树有一个入度为 0 的顶点，而其他所有顶点的入度都为 1，则称此有向树为**根树**。入度为 0 的顶点称为**树根**，出度为 0 的顶点称为**树叶**，树根和树叶以外的顶点称为**分支顶点**。一个顶点的**级**是从树根到该顶点的通路的长度。

例 12.5.4　如图 12.20 所示，图 12.20(a)和(b)表示的是同一棵根树，其中 v_1、v_2 和 v_3 是 1 级顶点，v_4、v_5 和 v_6 是 2 级顶点。

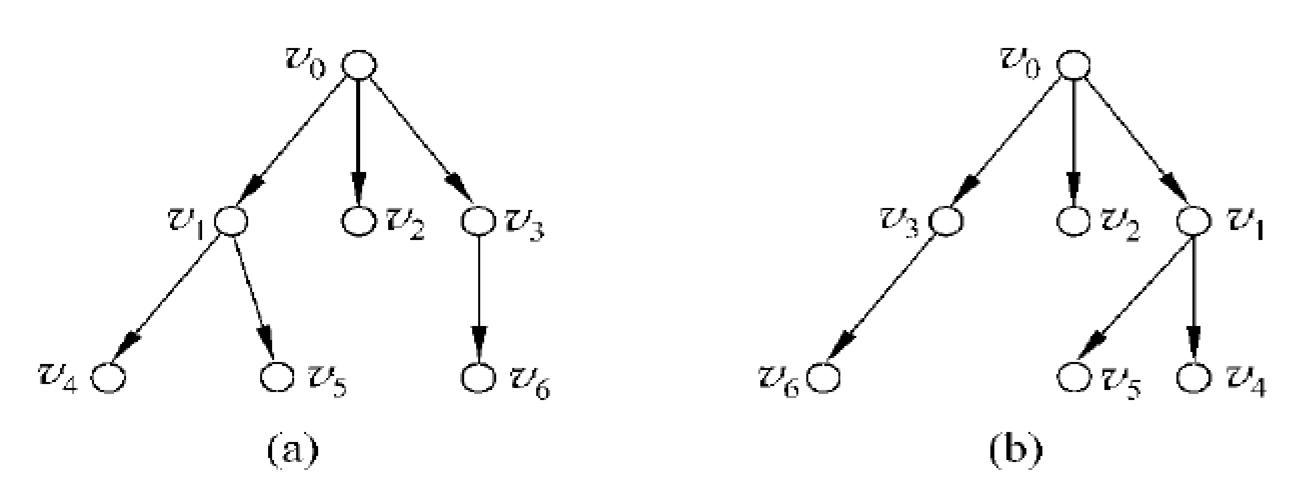


图 12.20　例 12.5.4 用图

定义 12.5.7　如果在有向树中规定了每一级上顶点的顺序，则这样的树称为有序树。

图 12.20 所示的(a)和(b)虽然是同一棵有向树，但却是两棵不同的有序树。

一棵根树可看成一个家族树。如果从 v_i 到 v_j 有一条边，那么称 v_i 为 v_j 的父亲，v_j 为 v_i 的儿子；如果从 v_i 到 v_j 有一条有向路，那么称 v_i 为 v_j 的祖先，v_j 为 v_i 的子孙；如果 v_i 和 v_j 有相同的父亲，那么称 v_i 和 v_j 为兄弟。

定义 12.5.8　T 为一棵根树，如果 T 的每个顶点的出度都不大于 m，则称 T 为 ***m* 元树**。如果 T 的每个顶点的出度都等于 m 或者 0，则称 T 为**完全 *m* 元树**。

在 m 元树中，应有最广泛的是**二元有序树**。这是由于二元有序树在计算机中易于处理，而且任何有序树或者森林都可以转化成二元树表示。在二元树的应用中，常常需要遍访树的每一个节点，也就是二元有序树的遍历问题。通常有 3 种方法：先

序遍历法、中序遍历法和后序遍历法。

先序遍历法 先访问二元有序树的树根 v_0；如果 v_0 有左儿子，则以先序遍历法遍访 v_0 的左子树（以 v_0 的左儿子为根的子树）；如果 v_0 有右儿子，则以先序遍历法遍访 v_0 的右子树（以 v_0 的右儿子为根的子树）。

中序遍历法 如果二元有序树的树根 v_0 有左儿子，则以中序遍历法遍访 v_0 的左子树；访问二元有序树的树根 v_0；如果树根 v_0 有右儿子，则以中序遍历法遍访 v_0 的右子树。

后序遍历法 如果二元有序树的树根 v_0 有左儿子，则以后序遍历法遍访 v_0 的左子树；如果树根 v_0 有右儿子，则以后序遍历法遍访 v_0 的右子树；访问二元有序树的树根 v_0。

例 12.5.5 对图 12.21 所示二元有序树的遍历结果如下：

(1) 先序遍历法：$a(b(cde)f)(igh)$

(2) 中序遍历法：$((dce)bf)a(gih)$

(3) 后序遍历法：$((dec)fb)(ghi)a$

图 12.21 例 12.5.5 用图

定理 12.5.5 二元树 T 有 n_0 个树叶，n_2 个出度为 2 的顶点，则 $n_2=n_0-1$。

证明：设出度为 1 的顶点数为 n_1，则 T 的顶点数 $n=n_0+n_1+n_2$。

又 T 的边数 $m=n-1$，$m=2n_2+n_1$，那么 $2n_2+n_1=n_2+n_1+n_0-1$，即

$$n_2 = n_0 - 1$$

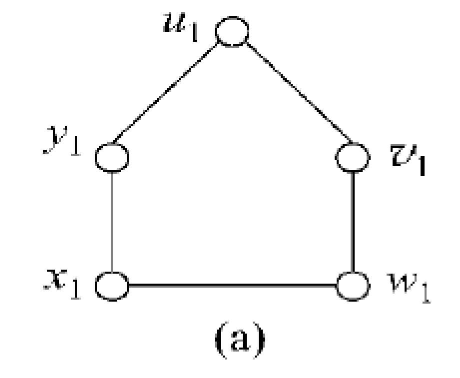
推论 12.5.1 完全二元树 T 有 n 个顶点，n_0 个树叶，则 $n=2n_0-1$。

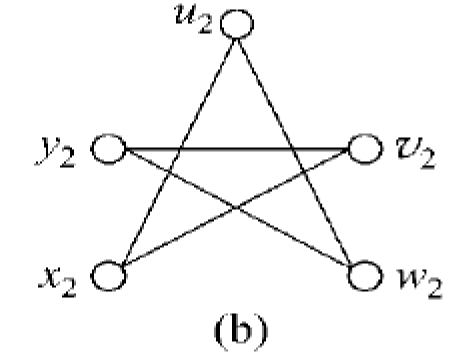
证明：设出度为 2 的顶点数为 n_2，则 $n=n_2+n_0$。由定理 12.5.5 可知，$n_2=n_0-1$，所以 $n=2n_0-1$。

12.6 图的同构

图的同构是图论中非常重要的一个问题，它在信息安全、图像、网络结构分析等领域中都有着广泛的应用。两个图之间的关系称为**同构**，直观上的理解就是两个图有着相同的形状，或者说两个图的顶点通过重新标号而形成相同的图。

例 12.6.1 在图 12.22 中，图 12.22(b)通过重新放置顶点，可以被重画为图 12.22(c)。因此图 12.22(a)、(b)的区别仅仅在于它们顶点的标号方式与图的画法，也就是说，它们有相同的结构。





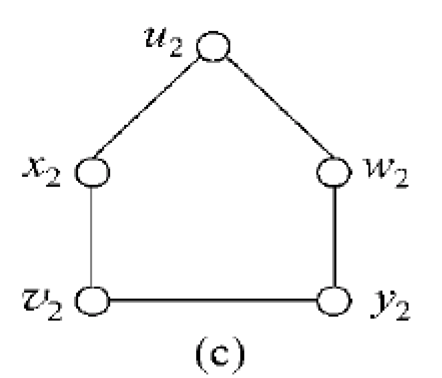


图 12.22 例 12.6.1 用图

定义 12.6.1　两个图 $G=(V,E)$和 $G'=(V',E')$称为是**同构的**(**isomorphic**)，如果存在一个双射 $f:V\to V'$，使得：$\{u,v\}\in E$ 当且仅当$\{f(u),f(v)\}\in E'$。也称 G'**同构于** G，记为 $G'\cong G$。

例 12.6.2　图 12.22 中，存在映射 $f:V_a\to V_b$，其中 $f(u_1)=u_2$，$f(v_1)=w_2$，$f(w_1)=y_2$，$f(x_1)=v_2$，$f(y_1)=x_2$。因此图 12.22(a)和图 12.22(b)是同构的。然而，图 12.23 中的两个图的形状很相似，但它们不是同构的。

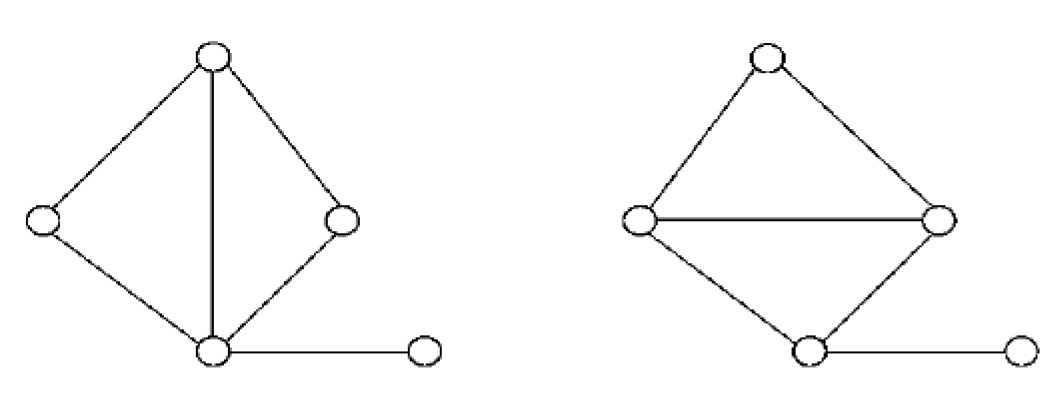


图 12.23　不同构树

定义 12.6.2　若图 G 和它的补图 $\overline{G}$ 是同构的，则称图 G 为**自补图**。

根据同构的定义，若两个图同构，则一定有：它们有相同的顶点数；有相同的边数；对应的顶点的度数相同。对于一个矩阵 $\boldsymbol{A}$，将它进行以下变换，第 i 行和第 j 行交换，然后第 i 列和第 j 列交换，这样的变换称为 $\boldsymbol{A}$ 的一次**对称变换**。12.3 节中讲到过图的矩阵表示，它们是计算机处理图的基础，其中的邻接矩阵可以用来很好地判断两个图之间的同构关系。

例 12.6.3　矩阵的一次对称变换表示如下：

$$\begin{pmatrix} a_{11} & \cdots & a_{1i} & \cdots & a_{1j} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots & & \vdots & & \vdots \\ a_{i1} & \cdots & a_{ii} & \cdots & a_{ij} & \cdots & a_{in} \\ \vdots & & \vdots & \ddots & \vdots & & \vdots \\ a_{j1} & \cdots & a_{ji} & \cdots & a_{jj} & \cdots & a_{jn} \\ \vdots & & \vdots & & \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{ni} & \cdots & a_{nj} & \cdots & a_{nn} \end{pmatrix} \to \begin{pmatrix} a_{11} & \cdots & a_{1j} & \cdots & a_{1i} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots & & \vdots & & \vdots \\ a_{j1} & \cdots & a_{jj} & \cdots & a_{ji} & \cdots & a_{jn} \\ \vdots & & \vdots & \ddots & \vdots & & \vdots \\ a_{i1} & \cdots & a_{ij} & \cdots & a_{ii} & \cdots & a_{in} \\ \vdots & & \vdots & & \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{nj} & \cdots & a_{ni} & \cdots & a_{nn} \end{pmatrix}$$

定理 12.6.1　图 $G_1=(V_1,E_1)$和图 $G_2=(V_2,E_2)$同构的充要条件是 G_1 对应的邻接矩阵 $\boldsymbol{A}_1$ 可以通过有限次的对称变换得到 G_2 对应的邻接矩阵 $\boldsymbol{A}_2$。

证明：必要性。假设两个图 G_1 和 G_2 同构，V_1 和 V_2 间存在一个双射 $f:V_1\to V_2$。记 $V_1=\{v_1,v_2,\cdots,v_n\}$，$V_2=\{w_1,w_2,\cdots,w_n\}$，则对 $\forall v_i\in V_1$ 都有 $f(v_i)=w_j(i,j\in[1,n])$。对 $\boldsymbol{A}_1$ 做以下系列变换，将每个 $v_i(i\in[1,n])$对应的行换到 w_j 对应的行(假定是第 k 行)，然后将 v_i 对应的列换到第 k 列，最终使得 $\boldsymbol{A}_1$ 的顶点顺序与 $\boldsymbol{A}_2$ 的顶点顺序相一致。

根据同构的定义，G_1 和 G_2 对应的顶点的邻接情况完全相同，也就是说经过系列变换后的 $\boldsymbol{A}_1$ 和 $\boldsymbol{A}_2$ 的对应行列是相同的，因此变换后的 $\boldsymbol{A}_1$ 和 $\boldsymbol{A}_2$ 完全相等。上面对 $\boldsymbol{A}_1$ 做的变换就是一系列的对称变换，即 $\boldsymbol{A}_1$ 通过有限次的对称变换可得到 $\boldsymbol{A}_2$。

充分性。假设图 G_1 的邻接矩阵 $\boldsymbol{A}_1$ 经过有限次的对称变换后可得到图 G_2 的邻

接矩阵 $\boldsymbol{A}_2$。记 $V_1=\{v_1,v_2,\cdots,v_n\}$，$V_2=\{w_1,w_2,\cdots,w_n\}$。对于 $\forall v_i\in V_1$，v_i 若对应变换后矩阵的第 j 行，那么就记 $f(v_i)=w_j$。而且这种对应关系是唯一的，因此 f：$V_1\to V_2$是一个双射。

对于 $\forall\{v_k,v_s\}\in E_1$，令 $f(v_k)=w_p$，$f(v_s)=w_q$，由于 A_1 中 v_k 行 v_s 列对应的项为 1，根据对称变换，那么在 $\boldsymbol{A}_2$ 中第 p 行第 q 列对应的项也应该为 1，即$\{f(v_k),f(v_s)\}\in E_2$。同理，若 $\forall\{v_k,v_s\}\notin E_1$，则$\{f(v_k),f(v_s)\}\notin E_2$。即$\{v_k,v_s\}\in E_1$，当且仅当$\{f(v_k),f(v_s)\}\in E_2$。因此 G_1 和 G_2 是同构的。证毕。

根据定理 12.6.1 可以很自然地得到下面判断自补图的定理。

定理 12.6.2 一个图 G 为自补图的充要条件是 G 的邻接矩阵 $\boldsymbol{A}$ 通过有限次的对称变换得到 $\overline{G}$ 的邻接矩阵 $\overline{\boldsymbol{A}}$。

图同构是作用在图集合上的一个等价关系，因此可以把图集合划分成一些等价类(子集)，称之为**同构类(isomorphism class)**。属于同一个同构类的任意两个图是同构的，属于不同同构类的任意两个图是不同构的。

判断两个图是否同构的问题称为图同构问题。除了图同构本身在实践中的重要性，它在计算复杂性理论中也有着重要的研究价值，它属于 NP 问题，但是无法判断其属于 P 问题或 NP 完全问题，这样的问题在 NP 中占的比例比较少。有些学者提出了 GI (Graph Isomorphism)问题类，来表示那些可以在多项时间内归约到图同构判定的问题。

12.7 应用举例

图论方法与技术在信息安全领域有着广泛而深入的应用，本节将重点介绍图同构问题在构建零知识证明系统中的应用以及三染色问题在计算复杂性中的应用。

12.7.1 基于同构图的零知识证明系统

零知识证明系统在现代密码学中处于核心位置，并且是定义和证明各种密码方案安全性的广为接受的方法。简单地说，零知识证明系统允许证明者向验证者证实一个论断，但是却不泄露任何(验证者在多项式时间内计算得不到的)知识。下面给出同构图的完备零知识证明的具体构造。

算法 12.7.1 同构图的完备零知识证明

- 公共输入：两个图 $G_1=(V_1,E_1)$和 $G_2=(V_2,E_2)$。
- 假设证明者 P 知道 G_1 和 G_2 同构，令 ϕ 是 G_1 到 G_2 的同构映射，即 ϕ：$V_1\to V_2$ 是双射，且$\{u,v\}\in E_1$ 当且仅当$\{\phi(u),\phi(v)\}\in E_2$。
- 下面的步骤将使验证者 V 相信 P 的知识。

(1) 证明者 P 随机置换 G_2 产生另一个同构图 $H=(V_2,F)$，其中 $F=\{(\pi(u),\pi(v))$：$(u,v)\in E_2\}$，π 是 G_2 到 H 的同构映射。P 将图 H 发给验证者 V(注：因为 P 知道 G_2 和 H 同构，她也就知道 G_1 和 H 同构。但

对其他人来说，发现 G_1 和 H 或 G_2 和 H 之间同构与发现 G_1 和 G_2 之间同构一样困难)。

(2) 接收到证明者 P 发送的图 H，验证者 V 均匀选取 $\sigma \in \{1,2\}$，然后将 σ 发给证明者，让它给出 H 和 G_σ 间的同构映射。

(3) 若证明者 P 从 V 收到的 $\sigma=2$，则 P 将 π 发送给 V；否则，P 发送 $\pi\circ\phi(\pi\circ\phi(v)=\pi(\phi(v)))$。

(4) 若验证者 V 从 P 接收到的消息 ψ 是 G' 和 G_σ 间的同构映射，则 V 输出 1(接受输入)；否则输出 0(拒绝输入)。

(5) P 和 V 重复第(1)到(4)步 n 次。

上述构造算法具有以下 3 个属性：

(1) **完备性**。假设证明者 P 是诚实的参与者，严格按照上述步骤执行，验证者 V 总会接受输入；

(2) **有效性**。这个构造每运行一轮，P 都有 $\frac{1}{2}$ 的概率猜中验证者 V 在第(2)步中会要求她执行哪一个证明，从而对 V 进行欺骗，重复运行 n 轮后，P 成功欺骗的概率是 $\frac{1}{2^n}$；

(3) **零知识性**。证明者 P 在每一轮构造中都产生一个新图 H，运行 n 轮后，验证者 V 仅得到图 G_1 或 G_2 的一些随机同构副本，没有得到任何有用的信息以帮助他了解 G_1 和 G_2 之间的同构性。

把满足上述 3 个属性的算法或协议称为一个完备的交互零知识证明系统。因此，可证明以下定理。

定理 12.7.1　算法 12.7.1 构造出一完备的交互零知识证明系统。

12.7.2　三染色问题及其应用

图论在计算复杂性中也有着很好的应用。在关于 NP 完全问题(NP-Complete，简称 NPC)的一些证明中经常会用到图论的知识；图论中也有许多问题都是属于 NPC 的，三染色问题就是其中的一个。

定义 12.7.1　对于简单图 G，如果可以用 3 种颜色对 G 的顶点染色，使得任意两个相邻顶点具有的颜色不同，则称图 G 是可**三染色**的。

定理 12.7.2　三染色问题属于 NPC 问题。

证明：由于 NAESAT 是已知的 NPC 问题，只需构造一个从 NAESAT 到三染色问题的归约。NAESAT 是指对给定的一个子句集 $C_1,\cdots,C_m$，每个子句由 3 个文字组成，涉及的变量有 $x_1,x_2,\cdots,x_n$，问是否存在这样的一个真值指派，要求每个子句的文字不全为 true 或者不全为 false。

将构造一个图 G，使得它能够被 $\{0,1,2\}$3 种颜色染色当且仅当所有子句取不同的值。三角形是要用到的一个重要组件，它的 3 个顶点必须用 3 种不同颜色来染色。对每个变量 x_i 都存在一个三角形 $[a,x_i,\neg x_i]$ 与之对应；所有这些三角形共享一个

顶点 a,如图 12.24 中上面的几个三角形所示。每个子句 C_i 都可以用一个三角形 $[C_{i1}, C_{i2}, C_{i3}]$ 表示,如图 12.24 下面的三角形所示。最后,在 C_i 的第 j 个文字与 C_{ij} 间存在着一条边,这样就完成了图 G 的构造。下面证明图 G 可以被{0,1,2}染色当且仅当给定的 NAESAT 实例可满足。

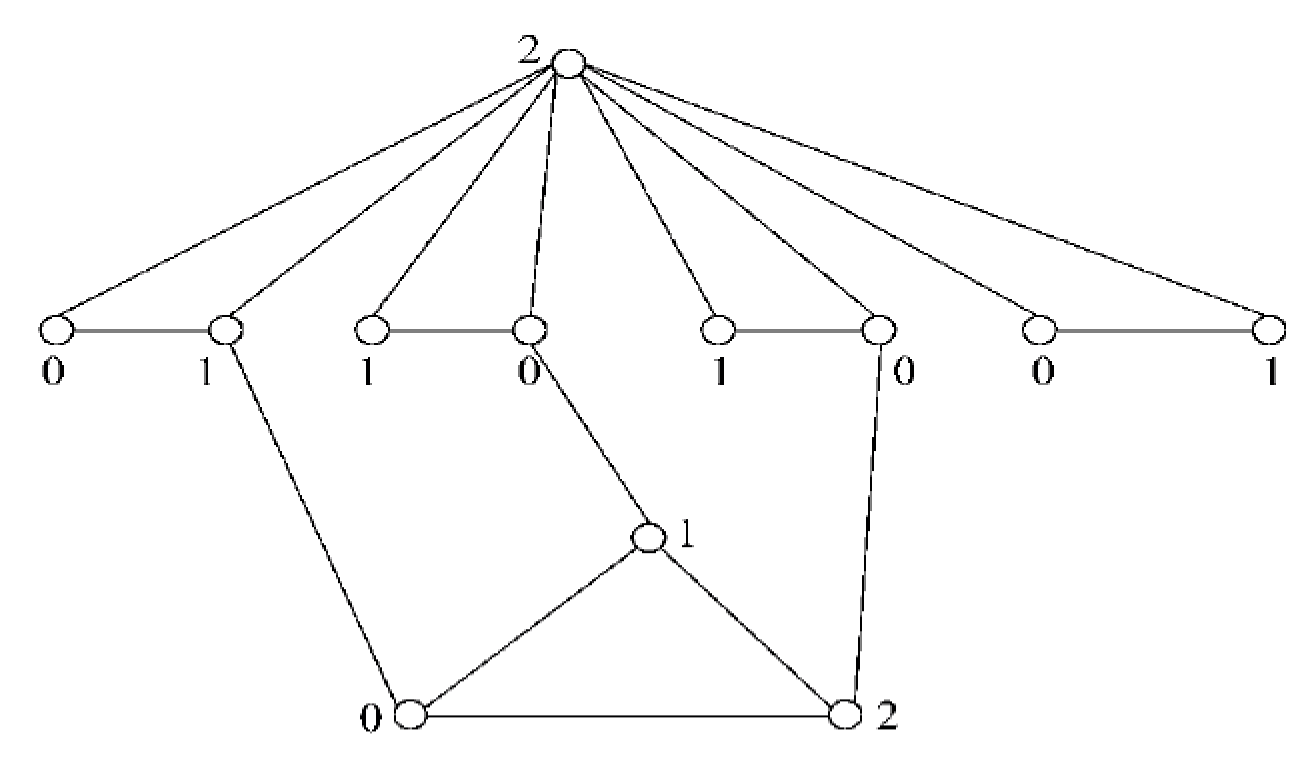


图 12.24　三染色问题图示

假设图 G 能够被三染色。通过改变颜色可以保证节点 a 一定是被颜色 2 染色,并且对每个 i, x_i 和 $\neg x_i$ 用 1 和 0 来染色。若 x_i 被 1 染色,则令对应的变量取 true;否则取 false。对于子句对应的三角形,若一个子句中所有文字都为 true,则对应的三角形就不能被染色;同理子句中所有文字也不可能都为 false。也就是说,对应的 NAESAT 实例可满足。

假设 NAESAT 的真值指派存在。用 2 对 a 染色,变量对应的顶点根据它的真值来染色,若为 true,则用 1 染色;若为 false,则用 0 染色。子句对应的三角形用下面的方法染色:找到三角形中两个取值相反的文字,对应的顶点用颜色 0 和 1 来染色,若文字取 true,则用 0 染色;若文字取 false,则用 1 染色,第三个顶点用 2 染色。这样就用{0,1,2}完成了对图 G 的三染色。证毕。

事实上,所有的 NP 语言都可以用来构建零知识证明系统,因此也可以构建一个基于三染色问题的零知识证明系统。

12.8　注记

本章重点介绍了一些在信息安全研究中常用的图论方法和技术,同时用典型实例阐述了图论方法和技术在信息安全领域中的应用。图论是一门发展比较成熟的学科,有着丰富的研究成果和广泛的应用。有很多图论方面的著作,如文献[1]~[4],尤其文献[1]和[2]是非常经典的著作。关于图论在信息安全领域中的应用,图同构是其中很重要的一个方面,如文献[5]、[6]。图论在密码学中的应用可参阅文献[7],图论在计算复杂性中的应用可参阅文献[8]。图论方面的一些最新进展可在《Jounal of Graph Theory》上找到,这也是图论研究领域的顶级刊物。

参 考 文 献

[1] Bond J A,Murty U S R 著.吴望名,李念祖译.图论及其应用.北京：科学出版社,1984
[2] Reinhard Diestel. Graph Theory(3rd edition). New York: Springer-Verlag Heidelberg,2005
[3] 孙惠泉.图论及其应用.北京：科学出版社,2004
[4] 洪帆.离散数学基础.武汉：华中科技大学出版社,1995
[5] Oded Goldreich,Silvio Micali,Avi Wigderson. Proofs that Yield Nothing but Their Validity or All Language in NP Have Zero-Knowledge Proof Systems, Journal of the Association for Computing Mechinery, Vol. 38, No. 1, 691-729, 1991
[6] Foggia P, Sansone C, Vento M. A performance comparison of five algorithms for graph isomorphism, In Proceedings of the 3rd IAPR-TC15 Workshop on Graph-based Representationsin Pattern Recognition, Ischia (Italy), May 2001
[7] Oded Goldreich. Foundations of Cryptography Basic Tools, Cambridge University Press, 2001
[8] Christos H. Papadimitriou. Computational Complexity. 北京：清华大学出版社,2004

第13章　数理逻辑方法与技术

在现代数学和计算机科学中，演算是最为普遍的科学行为。所谓演算就是利用一些符号，使用一组演算规则，进行逻辑的推演。例如，在微积分学中，人们规定一系列符号，如 x、y、z 等表示实数，根据极限的概念使用这些符号推导出一般的微分或积分公式。这种使用一定的符号，以及一定的规则构成公式，就形成了一种语言。再连同一些推演规则一起，形成了一个推演系统。一个积分公式在数学家的思想中有着确定的意义。而同一个公式，在不同的解释下，有着不同的正确性。例如，下面的公式：

$$\text{存在一个 } x, \quad \text{使得 } x^2 = 2$$

在有理数范围内解释，这个公式是不正确的。而在实数范围内，这个公式就是正确的。这种对于符号赋予的意义（或者解释），在逻辑中叫做语义。

逻辑系统的产生和应用最早始于哲学范畴，后来人们使用符号代替自然语言，进行推演、论证，从而形成数理逻辑系统。典型的逻辑系统由语言、推演系统及模型论语义3个要素组成。

语言是所讨论问题的载体，用以表达一些论断或者命题和公式。数理逻辑中，语言是一个无限字母表上符号串的集合，通常称为形式语言。构成逻辑公式的规则，称为逻辑的语法。

一个公式的正确与否是与这个公式在什么范围中解释，以及如何解释是有关的。这种在一定范围中的解释，就给每个公式确定了一定的意义，就是这个公式的语义。模型论是研究语法和语义关系的一个逻辑分支。

一个语言的推演系统是为了证明语言中公式或论断的真伪。为了证明一个公式是真的，从一些恒真的公式出发，运用一系列的推演规则，以及已经推演的结论，最终推导出这个公式。这些恒真的公式，就是公理。这些推演规则，就是系统的推演法则。研究有效的推演或证明方法的逻辑分支就是证明论。

数理逻辑在计算机科学乃至信息安全中有着广泛的运用。主要原因在于，逻辑是计算机科学的基础学科。形式化地表达一个问题，有助于利用计算机解决问题。要证明某些系统的可靠性，人们首先将这个系统用逻辑语言表达出来，再用逻辑的语言把需要验证的性质正确地表示出来，然后通过各种手段对其证明或验证。这是计算机科学中常用的手法，也毫不例外地是信息安全中所采用的方法。这些方法从信息安全发展之初就被广泛加以利用。并且可以预见，将来仍然是重要的方法之一。这一点从计算机和网络的访问控制到安全协议的设计与分析，都有重要体现。

本章主要介绍逻辑的基本概念和构成，介绍基本的逻辑系统：命题逻辑和一阶逻辑。目的是使读者了解和掌握基本的逻辑概念和组成元素。掌握这些必要的逻辑知识是进一步开展形式化方法在信息安全方面应用研究工作的前提。此后，介绍在

安全协议分析中有着重要影响的 SVO 逻辑的语法、语义及该逻辑的公理系统，并运用该逻辑系统完整地分析一个密钥协商协议。目的是使读者初步了解安全协议分析的必要步骤和方法。

13.1　命题逻辑

13.1.1　命题逻辑的语法

命题逻辑以命题为研究目标，因此命题逻辑的语言中以命题变元为最小的研究单位，称为原子公式。命题逻辑中有两个真值符号：$\top$(真)和$\bot$(假)。

定义 13.1.1　命题逻辑的**字母表**由下列组成：

(1) 一个可数无限的字母集合，表示命题变元：$A,B,C,\cdots,Z,A_1,A_2,\cdots$。

(2) 逻辑连接符号：$\neg$(非)，$\wedge$(且)，$\vee$(或)，$\rightarrow$(蕴含)，$\leftrightarrow$(等价)。

(3) 常量：$\bot$和$\top$。

(4) 辅助符号："(" 左括号、")"右括号。

字母表中的一些符号排列在一起，形成一个符号串。一个逻辑系统的公式是按照一定的规则形成的符号串。命题逻辑中的公式，由常量$\bot$和$\top$以及命题变元，通过连接符号适当的连接而成。

定义 13.1.2　**命题逻辑的公式**递归定义如下：

(1) 常量$\bot$、$\top$和每一个命题变元是公式。它们称为原子公式，把这个集合记为 SP(简单命题公式)。

(2) 如果 F、F_1、F_2 是公式，则 $\neg F$、$F_1\wedge F_2$、$F_1\vee F_2$、$F_1\rightarrow F_2$、$F_1\leftrightarrow F_2$ 都是公式。对于命题变元 P，命题公式 P 和 $\neg P$ 均称为文字。这时 F、F_1 和 F_2 称为对应公式的子公式。

(3) 任何一个公式都由上述方式构成。

上述形式生成的所有命题公式的集合记为 PROP。一般把单个命题变元以及命题常量$\bot$(即 SP 中的公式)称为简单命题或简单公式。而经过连接符号产生的命题称为复合命题或复合公式。

例 13.1.1　下面的字符串 Δ_1、Δ_2 和 Δ_3 是复合命题公式

$$\Delta_1:(P\rightarrow(Q\rightarrow R))\rightarrow((P\rightarrow Q)\rightarrow(P\rightarrow R))$$

这个公式含有 3 个原子公式：P,Q,R 共有下面 9 个子公式：

P、Q、R、$Q\rightarrow R$、$P\rightarrow(Q\rightarrow R)$、$P\rightarrow R$、$P\rightarrow Q$、$(P\rightarrow Q)\rightarrow(P\rightarrow R)$、$\Delta_1$

除 Δ_1 外，其余的是真子公式。

$$\Delta_2:P\rightarrow(Q\rightarrow P)$$

这个公式具有 P、Q 两个原子公式。共有 4 个子公式：P、Q、$Q\rightarrow P$、Δ_2

其中前 3 个是真子公式。

$$\Delta_3:((\neg P)\rightarrow(\neg Q))\rightarrow(Q\rightarrow A)$$

这个公式共有

$$P、Q、\neg P、\neg Q、(\neg P)\rightarrow(\neg Q)、Q\rightarrow P、\Delta_3$$

6 个子公式。

为了避免公式中括号的烦赘，一般定义以下的从高到低的连接符的优先级：

$$\neg,\ \wedge,\ \vee,\ \rightarrow,\leftrightarrow$$

其中连接词→、↔向右结合。根据这个规则，Δ_3 就可以简化为

$$\Delta_3:(\neg P\rightarrow\neg Q)\rightarrow(Q\rightarrow P)$$

但是公式 $\neg P\rightarrow\neg Q\rightarrow Q\rightarrow P$ 等价于 $\neg P\rightarrow(\neg Q\rightarrow(Q\rightarrow P))$，而与 Δ_3 不等价。读者会发现，有许多不同的公式，有着相同的意义。即在命题逻辑的语义下，它们等价。

13.1.2　命题逻辑的语义

在实际生活中，一个命题就是一个论断，可以判断真假。比如，“太阳现在的表面温度很高”就可以判断为真命题。而命题：“太阳绕着地球转”就是一个假命题。尽管这个命题在几百年前曾被认为是一个真命题，但是它是一个假命题。虽然人们的认识水平可以改变，但是命题的真假是不能改变的，不能够随意确定的。再如，黎曼假设实际上是一个命题，尽管目前还不知道它的真假，但是它的真假值是一定的。通常把命题的真假值称为命题的真值。这样，在古典逻辑中，一个命题的真值可以为真，也可能是假，二者必居其一。

用命题语言中的⊥表示假命题，⊤表示真命题。命题变元可以表示任意的命题。所以命题的真值是集合 BOOLEAN=\{0,1\}。这个集合也称为布尔集合。其中 0 表示假，而 1 表示真。命题⊥的真值为 0。这种把命题的真值假设为真假两个值的逻辑，也叫二值逻辑。这里不涉及多值逻辑。

一个复合命题的真值可以通过其中的原子命题的真值来确定。也就是说，从公式 F、F_1 和 F_2 真值，就能够得到 $\neg F$、$(F_1\wedge F_2)$、$(F_1\vee F_2)$、$(F_1\rightarrow F_2)$、$(F_1\leftrightarrow F_2)$的真值。它们之间的关系可以列表如表 13.1 所示。

表 13.1　逻辑连接符的真值表

P	Q	$\neg P$	$P\wedge Q$	$P\vee Q$	$P\rightarrow Q$	$P\leftrightarrow Q$
0	0	1	0	0	1	1
0	1	1	0	1	1	0
1	0	0	0	1	0	0
1	1	0	1	1	1	1

这些关系可以解释如下：

$\neg P$ 的真值恰与 P 的真值相反；

$P\wedge Q$ 的真值为 1 当且仅当 P 和 Q 的真值都为 1；

$P\vee Q$ 的真值为 1 当且仅当 P 和 Q 至少有一个真值为 1；

$P\rightarrow Q$ 真值为 0 当且仅当 P 为假时，Q 为真；

$P\leftrightarrow Q$ 真值为 1 当且仅当 P 和 Q 具有相同的真值。

根据这个解释，当原子命题变元的真值确定后，所有复合命题的真值也就随之确

定了。

定义 13.1.3　把 SP 到 BOOLEAN 的任何一个映射称为一个**赋值**。

任何一个赋值，都给每个命题变元指派了一个值。根据命题公式构成的定义，一个复合命题的真值就可以通过递归的方式，利用表 13.1 中连接符的真值解释，而最终确定。$\top$的赋值永远为 1，$\bot$的赋值为 0。

假设赋值 v 满足 $v(P)=1$，$v(Q)=0$，$v(R)=1$，那么公式

$$\Delta: P \land Q \rightarrow \neg Q \lor R$$

在 v 下的真值就是：$v(\Delta)=1\land 0\rightarrow \neg 0\lor 1=0\rightarrow 1=1$

任何一个命题公式都是有限长的符号串，因而其中含有的命题变元的个数有限。当这些命题变元的赋值确定后，这个命题的真值就唯一确定了。所以，对于一个命题公式，只有有限多个能够影响其真值的赋值。把一个命题公式在所有相关赋值下的真值总结到一起，形成它的真值表。如上面公式 Δ 的真值表如表 13.2 所示。

表 13.2　真值表

P	0	0	0	0	1	1	1	1
Q	0	0	1	1	0	0	1	1
R	0	1	0	1	0	1	0	1
Δ	1	1	1	1	1	1	0	1

从上述表中可以看出，由于公式中含有 3 个命题变元，所以真值表中涉及 2^3 个赋值。一般地，如果一个公式含有 m 个命题变元，那么这个公式的真值表中将涉及 2^m 个赋值。上面的真值表中，在某个赋值 v 下，公式的真值是 1；有些赋值下，公式的真值是 0。但是有些公式的真值表中，所有的赋值都是 1。

定义 13.1.4　一个公式 F 在某个赋值 v 下真值为 1，则说这个公式是可满足的，记为 v　F。如果一个公式 F 在任何赋值下都是 1，这个公式称为有效的，或者称为重言式，记为　F。如果一个公式 F 在任何赋值下，真值为 0，则说这个公式在这个赋值下不满足，记为 v　F。如果一个公式在任意赋值下都不满足，赋值满足则称为是一个矛盾或者不可满足的。

显然对于任何一个重言式 Δ，公式 $\neg\Delta$ 是不可满足的。可以验证，上节中给出的 3 个公式：Δ_1、Δ_2、Δ_3 都是重言式。

公式的可满足性在逻辑推演和计算机科学中有着重要的意义。例如，模型检测就是可满足性判定的一种方法。在工业设计，计算机软件可靠性分析中，有着重要的应用。

一个所谓的判定问题，就是给定问题的一个输入，回答为“是”或“否”的问题。例如，命题逻辑的可满足性问题，就是给定一个命题公式，判定它是否可以满足。这个问题就是一个判定问题。一个判定问题是可解的，意思是存在一个算法，对于这个问题的任何一个实例作为输入，这个算法都会在有限步内停止，给出“是”或者“否”的回答。如果不存在这样的算法，那么这个判定问题是不可解的。

命题逻辑的可满足性问题是一个可解的问题，或者说是一个可判定问题。这个

问题的一个算法是：给定一个命题公式，列出其真值表，如果有某一个赋值下，公式的真值为 1，则是可满足的。这个过程可以在有限多步下完成，因而是一个判定过程。

但是这个判定过程的效率是非常低的。一个含有 m 个原子公式的公式，要考虑 2^m 个赋值。当原子公式的个数较大时，这是一个不可接受的判定过程。

通过研究公式的构成，从而得到有效的判定过程，这是数理逻辑研究的一个主要的内容。

13.1.3 语义推论与语义等价

定义 13.1.5 两个命题公式 F、G，对于任意的赋值 v 满足：如果 v F 则 v G，则说 G 是 F 的一个推论，记为 F G。如果 F 是 G 的推论，并且 G 是 F 的推论，则说 F 和 G 是等价的，记为 $F \Leftrightarrow G$。如果 Γ 是一个公式集合，并且任意赋值 v 满足：如果对于 Γ 中任意公式 F，v F，那么一定有 v G，则说 G 是 Γ 的一个推论，记为 Γ G。

从定义中可以看出，两个公式等价，当且仅当两个公式的真值在任何赋值下相同。特别，任何重言公式等价于 $\top$，任何矛盾公式等价于 $\bot$。关于推论有下面的性质：F G 当且仅当 $F \to G$ 是重言公式。这可以通过真值表证明。下面是几个常见的等价公式，它们的证明可以通过真值表得到。

(1) $F \to G \Leftrightarrow \neg F \vee G$。

(2) $F \vee (G \wedge H) \Leftrightarrow (F \vee G) \wedge (F \vee H)$ （分配律）。

(3) $F \wedge (G \vee H) \Leftrightarrow (F \wedge G) \vee (F \wedge H)$ （分配律）。

(4) $\neg (F \wedge G) \Leftrightarrow \neg F \vee \neg G$ （De Morgan 法则）。

(5) $\neg (F \vee G) \Leftrightarrow \neg F \wedge \neg G$ （De Morgan 法则）。

(6) $E \vee \neg E \Leftrightarrow \top$。

在公式的等价证明中，有一个等价代换的法则：将一个公式的某个子公式代换为与这个子公式等价的公式，得到的公式与原来的公式等价。

应用上述公式，利用等价代换的法则，可以方便地证明许多公式的等价性。

(7) $((C \wedge D) \vee A) \wedge ((C \wedge D) \vee B) \wedge (E \vee \neg E) \Leftrightarrow (A \wedge B) \vee (C \wedge D)$。

证明：记左边的公式为 φ，那么 φ 的最末一个子公式为 $E \vee \neg E$ 等价于 $\top$，由此 φ 等价于 $((C \wedge D) \vee A) \wedge ((C \wedge D) \vee B)$，根据分配律，它等价于 $(C \wedge D) \vee (A \wedge B)$ 等价于右边的公式。

13.1.4 命题逻辑推演系统

一个逻辑的推演系统由一些推演规则和一些公理组成。由这些公理，通过推演规则得到的公式，称为一个**定理**。从公理推导出一个定理，可能要经过一系列的推导，得到一系列的公式。把这些公式按顺序罗列起来就形成一个公式序列，称为定理的一个形式证明。在表 13.3 中列出常用的推导规则：其中 Γ 代表一集命题公式。用 Γ G 表示 G 可以由 Γ 推导出。如果 Γ 是空集，则记做 G。

表 13.3　命题逻辑推演法则

$\dfrac{G\text{ 在 }\Gamma\text{ 中}}{\Gamma \quad G}$（假设）　$\dfrac{\Gamma \quad G\text{ 且 }\Gamma\subseteq\Gamma'}{\Gamma' \quad G}$（单调性）　$\dfrac{\Gamma \quad G}{\Gamma \quad \neg\neg G}$（双重否定）
$\dfrac{\Gamma \quad F,\Gamma \quad G}{\Gamma \quad F\wedge G}$（引入∧）　$\dfrac{\Gamma \quad F\wedge G}{\Gamma \quad F}$（消去∧）
$\dfrac{\Gamma \quad F}{\Gamma \quad F\vee G}$（引入∨）　$\dfrac{\Gamma \quad F\vee G,\Gamma\cup\{F\} \quad H,\Gamma\cup\{G\} \quad H}{\Gamma \quad H}$（消去∨）
$\dfrac{\Gamma \quad F\wedge G}{\Gamma \quad G\wedge F}$（∧的对称性）　$\dfrac{\Gamma \quad F\vee G}{\Gamma \quad G\vee F}$（∨的对称性）
$\dfrac{\Gamma\cup\{F\} \quad G}{\Gamma \quad F\rightarrow G}$（引入→）　$\dfrac{\Gamma \quad F\rightarrow G,\Gamma \quad F}{\Gamma \quad G}$（消去→）
$\dfrac{\Gamma \quad (F\wedge G)\wedge H}{\Gamma \quad F\wedge G\wedge H}$（∧的括号规则）　$\dfrac{\Gamma \quad (F\vee G)\vee H}{\Gamma \quad F\vee G\vee H}$（∨ 的括号规则）
$\Gamma \quad F\vee G$　当且仅当　$\Gamma \quad \neg(\neg F\wedge\neg G)$　（∨的定义）
$\Gamma \quad F\rightarrow G$　当且仅当　$\Gamma \quad \neg F\vee G$　（→的定义）
$\Gamma \quad F\leftrightarrow G$　当且仅当　$\Gamma \quad F\rightarrow G$ 且 $\Gamma \quad G\rightarrow F$　（↔的定义）

表 13.3 所示是命题逻辑推演规则列表。表中的推演法则一般是$\dfrac{\text{前提}}{\text{结论}}$的形式。后面的括号中列出该法则的名称。下面就利用这些法则进行一些公式的证明。

例 13.1.2　证明下列命题：

（1）　$(P\rightarrow(Q\rightarrow R))\rightarrow((P\rightarrow Q)\rightarrow(P\rightarrow R))$；

（2）　$P\rightarrow(Q\rightarrow P)$；

（3）　$(\neg P\rightarrow\neg Q)\rightarrow Q\rightarrow A$。

细心的读者会发现，上述证明的公式正好是例 13.1.1 中的 Δ_1、Δ_2、Δ_3，即空集可以推演出 Δ_1、Δ_2、Δ_3。表 13.4 只给出 Δ_1的推演过程。

表 13.4　Δ_1的推演过程

推 演 过 程	论　　据
1. $\{P\rightarrow(Q\rightarrow R),P\rightarrow Q,P\} \quad P$	假设
2. $\{P\rightarrow(Q\rightarrow R),P\rightarrow Q,P\} \quad P\rightarrow Q$	假设
3. $\{P\rightarrow(Q\rightarrow R),P\rightarrow Q,P\} \quad Q$	1,2,消去→
4. $\{P\rightarrow(Q\rightarrow R),P\rightarrow Q,P\} \quad P\rightarrow(Q\rightarrow R)$	假设
5. $\{P\rightarrow(Q\rightarrow R),P\rightarrow Q,P\} \quad (Q\rightarrow R)$	1,4,消去→
6. $\{P\rightarrow(Q\rightarrow R),P\rightarrow Q,P\} \quad R$	3,5,消去→
7. $\{P\rightarrow(Q\rightarrow R),P\rightarrow Q\} \quad P\rightarrow R$	6,引入→
8. $\{P\rightarrow(Q\rightarrow R)\} \quad (P\rightarrow Q)\rightarrow(P\rightarrow R)$	7,引入→
9. $(P\rightarrow(Q\rightarrow R))\rightarrow((P\rightarrow Q)\rightarrow(P\rightarrow R))$	8,引入→

上述的证明过程的每一步，都可以清楚地用表 13.3 中的规则验证。实际上还有许多推演法则可供推演使用。利用表中给出的法则可以证明更多的推演法则：

例 13.1.3 下面是另外一些推演法则，只给出(1)的证明，如表 13.5 所示。

表 13.5 例 13.1.3 中(1)的证明

规则(1)的推演过程	论　据
1. $\Gamma\cup\{F\}\quad G$	前提
2. $\Gamma\cup\{F\}\quad \neg\neg G$	对 1 实行双重否定
3. $\Gamma\quad F\to\neg\neg G$	2，引入→
4. $\Gamma\quad \neg F\vee\neg\neg G$	3，→的定义
5. $\Gamma\quad \neg\neg G\vee\neg F$	4，∨的对称性
6. $\Gamma\quad \neg G\to\neg F$	5，引入→
7. $\Gamma\cup\{\neg G\}\quad \neg G\to\neg F$	6，单调性
8. $\Gamma\cup\{\neg G\}\quad \neg G$	假设
9. $\Gamma\cup\{\neg G\}\quad \neg F$	7，8，消去→

(1)（逆否法则） $\dfrac{\Gamma\cup\{F\}\quad G}{\Gamma\cup\{\neg G\}\quad \neg F}$。

(2)（矛盾法则） $\dfrac{\Gamma\quad F\wedge\neg F}{\Gamma\quad G}$。

(3)（重言式法则） $\dfrac{}{\Gamma\quad F\vee\neg F}$。

(4)（矛盾证明法则） $\dfrac{\Gamma\cup\{F\}\quad G,\Gamma\cup\{F\}\quad \neg G}{\Gamma\quad \neg F}$。

(5)（逐类证明法则） $\dfrac{\Gamma\cup\{F\}\quad G,\Gamma\cup\{\neg F\}\quad G}{\Gamma\quad G}$。

尽管给出了如此多的推演法则，但是仍然有许多常用的法则没有包括进来。比如，分配律、DeMorgan 律等。但是，它们都可以由表 13.3 中的规则推演得到。

在逻辑的推演系统中，一个最重要的问题就是，推演系统的**可靠性**。所谓推演系统的可靠性，就是一个公式可以由一个集合推演出来，那么这个公式一定是这个集合的一个推论。下面证明命题推演系统是可靠的。

定理 13.1.1 如果 $\Gamma\quad F$ 则 $\Gamma\quad F$。

证明： 如果 $\Gamma\quad F$，那么一定存在利用表 13.3 中的法则进行的一系列的推演，最终得到 $\Gamma\quad F$。我们只需验证每一步都是可靠的推演即可。实际上只需验证每条法则的可靠性即可。这里只验证表 13.3 中的两条规则的可靠性，其余的留给读者。

(1) 证明 $\dfrac{\Gamma\quad F\vee G,\Gamma\cup\{F\}\quad H,\Gamma\cup\{G\}\quad H}{\Gamma\quad H}$（消去∨）的可靠性。

证明： 由于前提 $\Gamma\quad F\vee G,\Gamma\cup\{F\}\quad H,\Gamma\cup\{G\}\quad H$ 意味着

$\Gamma\quad F\vee G,\Gamma\cup\{F\}\quad H,\Gamma\cup\{G\}\quad H$，证明 $\Gamma\quad H$。对于任意赋值 v，如果 $v\quad \Gamma$，由 $\Gamma\quad F\vee G$ 得知 $v\quad F$ 或者 $v\quad G$。

如果 $v\quad F$，则 $v\quad \Gamma\cup\{F\}$，再由 $\Gamma\cup\{F\}\quad H$ 知 $v\quad H$。

如果 $v\quad G$，则 $v\quad \Gamma\cup\{G\}$，再由 $\Gamma\cup\{G\}\quad H$ 知 $v\quad H$。

总之有 $v\quad H$。再由 v 的任意性知 $\Gamma\quad H$。

（2）证明$\frac{\Gamma \cup \{F\}\ \ G}{\Gamma\ \ F \to G}$（引入→）的可靠性。

证明： 前提假设 $\Gamma \cup \{F\}\ \ G$，证明 $\Gamma\ \ F \to G$。设赋值 v 满足 Γ。如果进一步有 $v\ \ F$，则由 $\Gamma \cup \{F\}\ \ G$ 知道，$v\ \ G$。如果 $v\ \ F$，根据→的语义解释一定有 $v\ \ F \to G$。将这个过程列真值表如表 13.6 所示。

表 13.6　真值表

$v(\Gamma)$	$v(F)$	$v(G)$	$v(F \to G)$
1	1	1	1
	0	*	1

表 13.6 中的 * 可以是 0 或者 1。由于 v 是任意的赋值，说明 $\Gamma\ \ F \to G$ 可靠性定理保证了，正确的命题推演得到的命题一定正确，这样就把语义的正确性和可证明性紧密联系起来。由此有下面的推论：

（1）空集推演出的公式一定是重言式。这是因为，如果 F 可以由空集推演得到，那么根据单调性，它可以由任何集合推演得到，即任何赋值满足 F。

（2）如果 $\neg F$ 可以被空集推演得到，那么 F 一定是矛盾式（即不可满足）。

13.2　一阶逻辑

这一节介绍一阶逻辑的基本概念。在上一节介绍的命题逻辑中，研究对象仅限于命题，至于命题是如何构成的则没有涉及。一阶逻辑则更进一步研究具有一定结构的命题，以及构成这些命题的个体形成的结构和所遵循的规律。因此，在一阶逻辑中，引入函数和谓词的概念。有了这些新的概念，一阶逻辑具有很强的表达能力。几乎可以表达一般数学中所有的命题和定理。尽管被称为一阶逻辑，并非表明它是最为简单的逻辑。由于本章目的是介绍用于表达安全性质的逻辑，并且由于篇幅的限制，在这里只介绍一般与其他逻辑共性的性质，也就是文献中经常涉及的性质。以便读者对于基础知识有一定的认识。

13.2.1　一阶逻辑的语法

定义 13.2.1　一阶逻辑语言的**字母表**含有下列符号：

（1）$v_1, v_2, \cdots$，（个体变元）；

（2）$\neg, \vee, \wedge, \to, \leftrightarrow$（非、或、与、蕴含、当且仅当）；

（3）$\forall, \exists$（全称量词、存在量词）；

（4）$=$（等于）；

（5）），（（括弧）；

（6）非逻辑符号：

① 对于任意 $n>0$，有一集 n 元的关系符号$\{R_1^n, R_2^n, \cdots\}$（可以是空集）。

② 对于任意 $n>0$，有一集 n 元的函数符号$\{f_1^n, f_2^n, \cdots\}$（可以是空集）。

③ 一集常量符号$\{c_1, c_2, \cdots\}$（可以是空集）。

习惯上，使用 $P,Q,R,\cdots$ 作为关系符号，关系符号也称为谓词符号。因此，一阶逻辑也称为谓词逻辑。使用 $f,g,h,\cdots$ 作为函数符号。而 $c,c_1,c_2,\cdots$ 表示常量符号。用 $x,y,z,\cdots$ 表示个体变元。

定义中，(1)～(5)包含的符号通称为逻辑符号，其中 ∀、∃ 称为量词符号。∀ 是全称量词，∃ 是存在量词。正是这些量词的使用，使得一阶逻辑具有强大的表达能力。

一个一阶语言由其非逻辑符号所决定。不同的非逻辑符号，决定了不同的语言。所以说一个一阶语言，往往只要表明其非逻辑符号。比如，群论的一阶语言为 $L_1=\{\circ,e\}$。其中 e 是一个常量符号，意欲表示群的单位元。符号∘是群的二元运算函数符号。域论的语言为 $L_1=\{+,\circ,0,1\}$。其中＋和∘是二元运算函数符号，分别表示加群和乘群的乘法运算。常量 0 表示环的单位元，1 表示乘法群的单位元。

在数学中，公式的项是一些变元、常量及函数的有意义的组合。在一阶逻辑中，有着同样的定义。

定义 13.2.2 一个项通过有限次的使用下述规则得到：

(1) 语言中的一个变元是一个项；

(2) 语言中的一个常量是一个项；

(3) 如果 $t_1,t_2,\cdots,t_n$ 是项，f 是语言的一个 n 元函数符，则 $f(t_1,t_2,\cdots,t_n)$ 是一个项。

公式是最为熟悉的数学表达式。利用项的定义，给出一阶逻辑的公式的定义。

定义 13.2.3 一个语言 L 的公式是通过有限次使用下述规则得到的符号串：

(1) 如果 t_1、t_2 是项，那么 $t_1=t_2$ 是一个公式；

(2) 如果 $t_1,t_2,\cdots,t_n$ 都是项，R 是语言的一个关系符号，那么 $R(t_1,t_2,\cdots,t_n)$ 是一个公式；

(3) 如果 φ 是一个公式，则 $\neg\varphi$ 是一个公式；

(4) 如果 φ、ψ 是两个公式，那么 $\varphi\wedge\psi$、$\varphi\vee\psi$、$\varphi\rightarrow\psi$、$\varphi\leftrightarrow\psi$ 都是公式；

(5) 如果 φ 是一个公式，x 是一个变元，则 $\forall x\varphi$ 和 $\exists x\varphi$ 都是公式。

其中仅仅通过(1)、(2)两项生成的公式称为原子公式。在(5)中，$\forall x\varphi$ 定义为 $\neg\exists x\neg\varphi$。这两个公式是可以互换的。

下面是一阶语言 $L_1=\{\circ,e\}$ 的一些公式，它们一起组成了群的公理：

$$\text{(G1)}\quad \forall x\forall y\forall z((x\circ y)\circ z=x\circ(y\circ z))$$

$$\text{(G2)}\quad \forall x(x\circ e=x)$$

$$\text{(G3)}\quad \forall x\exists y(x\circ y=e)$$

公式(G1)表示结合律，(G2)说明 e 是右单位元，(G3)说明每个元素都具有右逆元。

在某些公式中，个体变元都在量词 ∀、∃ 的辖域中，即这些公式没有自由变元。通常把这种没有自由变元的公式称为句子或语句。下面的公式就不是句子：

$$\forall x(R(x,y)\wedge\exists yQ(x,y))$$

这是因为个体变元 y 没有被任何量词约束。这种没有被任何量词约束的变元称为

自由变元。在 $\exists y Q(x,y)$ 中，变元 y 被存在量词所约束，因此不是自由变元。

13.2.2　一阶逻辑的语义

在命题逻辑中看到，一个命题的真值只有两种可能性，即真或假。而在一阶逻辑中，要复杂得多。

例如，公式 $\forall y \exists x f(x)=y$。在实数域中，假如 $f(x)=x^2$，那么当 $y<0$ 时，公式就不成立，故公式假。如果 $f(x)=x^3$，那么公式就真。再者，如果在整数环中，即使 $f(x)=x^3$，原公式也是假的。这说明，一阶逻辑中的公式的语义与个体变元的范围，或者称为论域（如实数或整数集合）有关。也与函数符号、关系符号的意义有关。

定义 13.2.4　一个语言 L 的**论域**，是一个集合 U_L 以及在这个集合中，L 的常量、函数符号及关系符号的解释。

L 的每个常量解释为 U_L 中的一个固定元素，n 元函数符号解释为这个集合上的一个 n 元函数，n 元关系符号解释为该集合上的一个 n 元的关系。

例子，语言 $L=\{0,1;+,\cdot;<\}$。这个语言中有两个常量符号、两个函数符号、一个关系符号。它的两个论域为：$R=\{\mathbb{R}|0,1;+,\cdot;<\}$ 和 $Z=\{\mathbb{Z}|0,1;+,\cdot;<\}$。其中$\mathbb{R}$ 是全体实数的集合，而$\mathbb{Z}$ 是整数集合。对于 0，1 的解释都是通常的整数 0 和 1；+和 · 是通常的加法和乘法；<是通常的小于关系。

对于语言 L，它的任意一个语句 φ 在 L 的一个论域 M 中或者为真或者为假。如果在 M 中为真，那么就说 M 是它的一个模型，记为 M　φ；否则说 M 不是 φ 的一个模型，记为 M　φ。

例如，对于语言 $L=\{0,1;+,\cdot;<\}$ 中的句子 $\varphi=\forall x \exists y(0<x\rightarrow y\cdot y=x)$，在论域 R 中是真的，即 R　φ；而 Z　φ。

对于语言 L 和它的任意一个语句 φ，如果 φ 在 L 的一个论域 M 中为真，就说 φ 是可满足的。如果 φ 在 L 的所有模型中都成立，那么 φ 是一个恒真式（或者称为重言式），记为　φ。如果在任意模型中都不成立，则说 φ 是一个矛盾式。

可满足性的定义可以推广到一般的公式上：假设 $\psi(x_1,x_2,\cdots,x_n)$ 是 L 的一个公式，那么 $\psi(x_1,x_2,\cdots,x_n)$ 在 M 中满足当且仅当 $\forall x_1 \forall x_2 \cdots \forall x_n \psi(x_1,x_2,\cdots,x_n)$ 在 M 中满足，即 M　$\forall x_1 \forall x_2 \cdots \forall x_n \psi(x_1,x_2,\cdots,x_n)$。

在数理逻辑中，模型论是一门研究语法和语义关系的数学分支，有着丰富的内容。其结果对于数学和计算机科学都有着积极的促进作用。在计算机程序验证和安全协议验证方面，模型检测是一个主要的手段。具体说来，假设一个性质可以用一个公式 φ 刻画，这个性质要满足的环境为一个模型 M。那么这个性质是否在这个环境中被满足，就是模型检测问题。即判定是否 M　φ。这里要强调的是，在计算机科学中，往往在不同的应用场景，采用不同的逻辑，如时态逻辑、模态逻辑等。在后面要介绍信念逻辑，就是为了适应安全协议的安全属性刻画的需要。

与命题逻辑一样，如果 ψ 的模型都是 φ 的模型，那么说 φ 是 ψ 的一个推论，记为 ψ　φ。如果进一步有 φ　ψ，则称 ψ 与 φ 等价，记为 $\psi\Leftrightarrow\varphi$。同样，　φ 表明 φ 是一个重言式。

13.2.3 一阶逻辑的推演系统

人们把命题逻辑的推演规则进行扩充，形成一阶逻辑的推演规则。其中 Γ 代表一集命题公式。用 $\Gamma \quad G$ 表示 G 可以由 Γ 推导出。如果 Γ 是空集，则记做 $\quad G$。一阶逻辑的推演规则是表 13.3 加上表 13.7 中的规则。

表 13.7 一阶逻辑推演规则(部分)

$\dfrac{\Gamma \quad \forall x\varphi(x)}{\Gamma \quad \neg\exists x\neg\varphi(x)}$ 且 $\dfrac{\Gamma \quad \neg\exists x\neg\varphi(x)}{\Gamma \quad \forall x\varphi(x)}$	(量词 ∀ 的定义)
$\dfrac{\Gamma \quad \varphi(t)}{\Gamma \quad \exists x\varphi(x)}$ (t 是一个项，x 是一个变元，并且 x 不在 φ 中约束出现)	(引入 ∃ 量词)
$\dfrac{\Gamma \quad \varphi(t_0)}{\Gamma \quad \forall x\varphi(x)}$ (x 是一个变元，不在 φ 中约束出现，t_0 是一个变元，或者一个不在 Γ 中出现的常量)	(引入 ∀)
$\dfrac{\Gamma \quad \varphi\to\theta}{\Gamma \quad \exists x\varphi(x)\to\exists x\theta}$ (量词 ∃ 的可分配性)	$\dfrac{\Gamma \quad \varphi\to\theta}{\Gamma \quad \forall x\varphi(x)\to\forall x\theta}$ (量词 ∀ 的可分配性)
$\dfrac{\Gamma \quad Q_1x(Q_2y\varphi)}{\Gamma \quad Q_1xQ_2y\varphi}$ (Q_1、Q_2 是量词)	(量词括弧法则)
$\dfrac{}{\Gamma \quad t\to t}$ (t 是一个项) (自反性)	$\dfrac{\Gamma \quad \varphi(t),\Gamma \quad t\to t'}{\Gamma \quad \varphi(t')}$ (t 和 t' 是项) (等式替换)

定义 13.2.5 一阶逻辑的形式证明是一系列 $X \quad \theta$ 形式的论断，这些论断都是经过上述表 13.3 和表 13.7 规则推演得到。如果 $\Gamma \quad \varphi$ 是这一系列论断中的一个，那么就说 φ 可以从 Γ 中推演得到。

由于形式证明需要对于每一步应用推演规则，所以一个命题的形式证明，要比通常的数学证明繁难。正如前述强调的一样，一个逻辑的推演系统，最重要的是其可靠性。如果一个推演系统的可靠性无法保证，这个推演系统就是一个无法使用的系统。对于一阶逻辑有以下定理。

定理 13.2.1 (可靠性)如果 $\Gamma \quad \varphi$ 则 $\Gamma \quad \varphi$。

直观上，这个定理说明，一个命题是形式可证明的，一定是语义正确的。要证明这个定理，就要对于每个推演规则进行说明。这里略去。因为着眼点在于逻辑的应用，因此给出三个推演过程，供读者体会，如表 13.8 至表 13.10 所示。

表 13.8 定理 13.2.1 推演过程(一)

推 演 过 程	论 据
1. $\Gamma \quad \forall x\varphi(x)$	假设
2. $\Gamma\cup\{\neg\varphi(t)\} \quad \forall x\varphi(x)$	对 1 用单调性
3. $\Gamma\cup\{\neg\varphi(t)\} \quad \neg\exists x\neg\varphi(x)$	对 2 用 ∀ 的定义
4. $\Gamma\cup\{\neg\varphi(t)\} \quad \neg\varphi(t)$	假设
5. $\Gamma\cup\{\neg\varphi(t)\} \quad \exists x\neg\varphi(x)$	对 4 进行 ∃ 量词引入
6. $\Gamma \quad \neg\neg\varphi(t)$	由 3,5 以及矛盾法则
7. $\Gamma \quad \varphi(t)$	对 6 用双重否定法则
8. $\Gamma \quad \exists x\varphi(x)$	对 7 用 ∃ 量词引入

表 13.9　定理 13.2.1 推演过程(二)

推演过程	论　据
1. $\Gamma \quad \varphi(x)$	假设
2. $\Gamma \quad (\varphi(x) \vee \neg(x=x))$	对 1 引入 $\vee$
3. $\Gamma \quad (\neg(x=x) \vee \varphi(x))$	对 2 用 $\vee$ 的对称性
4. $\Gamma \quad (x=x) \to \varphi(x))$	对 3 用→的定义
5. $\Gamma \quad \forall x(x=x) \to \forall x\varphi(x))$	对 4 进行 $\forall$ 量词分配
6. $\Gamma \quad c=c$	自反性法则
7. $\Gamma \quad \forall x(x=x)$	对 6 用引入 $\forall$ 量词
8. $\Gamma \quad \forall x\varphi(x)$	对 5,7 用→消去

表 13.10　定理 13.2.1 推演过程(三)

推演过程	论　据
1. $\Gamma \quad \exists x\varphi(x)$	假设
2. $\Gamma \cup \{\neg\varphi\} \quad \exists x\varphi(x)$	单调性
3. $\Gamma \cup \{\neg\varphi\} \quad \neg\varphi$	假设
4. $\Gamma \cup \{\neg\varphi\} \quad \forall x \neg\varphi(x)$	对 3 用前一个证明的结论
5. $\Gamma \cup \{\neg\varphi\} \quad \neg\exists x \neg\neg\varphi(x)$	对 4 用量词 $\forall$ 的定义
6. $\Gamma \cup \{\neg\varphi\} \quad \neg\exists x\varphi(x)$	对 5 用进行用双重否定
7. $\Gamma \quad \neg\neg\varphi(x)$	对 2,6 用矛盾法则
8. $\Gamma \quad \varphi(x)$	对 7 用双重否定

例 13.2.1　对于任何公式 $\varphi(x)$，有 $\forall x\varphi(x) \quad \exists x\varphi(x)$。

证明：记 $\Gamma=\{\forall x\varphi(x)\}$，欲证 $\Gamma \quad \exists x\varphi(x)$。在证明过程中，要用到上节例题中的矛盾法则：$\dfrac{\Gamma \cup \{G\} \quad \varphi, \Gamma \cup \{G\} \quad \neg\varphi}{\Gamma \quad \neg G}$。证明过程见表 13.8。

例 13.2.2　设 x 不在公式 $\varphi(x)$ 中自由出现，则公式 $\varphi(x)$，$\forall x\varphi(x)$，$\exists x\varphi(x)$ 是可证明等价的。

证明：(1) 先证明 $\varphi(x) \quad \forall x\varphi(x)$。令 $\Gamma=\{\varphi(x)\}$。证明过程见表 13.9。

(2) 再证明 $\exists x\varphi(x) \quad \varphi(x)$。记 $\Gamma=\{\exists x\varphi(x)\}$。证明过程见表 13.10。

(3) 由例 13.2.1 知道，$\forall x\varphi(x) \quad \exists x\varphi(x)$。

由上述(1)、(2)、(3)可知 $\varphi(x)$，$\forall x\varphi(x)$，$\exists x\varphi(x)$ 是证明等价的。

计算机科学中一个非常重要的概念称为可判定性。直观上说，一类问题是可判定的，意思是存在一个算法，对于这类问题中的**任何**一个问题用这个算法在有限步内得到这个问题的正确性判定。

具体到一阶逻辑，把一些句子的集合称为一个理论。对于一个理论 T，如果存在一个算法，使得任何句子 φ 都可以用这个算法在有限步内判定 φ 是否在这个理论中，或者由这个理论推演出，则说这个理论是可判定的。如果一个理论是不可判定的，意味着任何有限多个算法都无法实现判定过程。对于不可判定的问题，只能够寄希望于找到对于一部分问题的判定方法。例如，已经证明，认证协议是不可判定的，所以无法用一个算法实现认证协议的安全性判定过程。

13.3 SVO 逻辑

本节介绍一个用于安全协议分析的逻辑系统——SVO 逻辑。SVO 逻辑是由 Paul F. Syverson 和 Paul C. van Oorschot 在总结了 BAN 类逻辑的基础上提出的。BAN 逻辑是一个具有历史意义的逻辑。它是由 Burrows、Abadi 及 Needham 提出的一个专门用于认证协议安全性证明的逻辑。作为第一个专门为安全协议证明设计的逻辑系统，它的出现引起了学术界极大的兴趣。也标志着逻辑系统在安全协议证明的应用方面的开端，因而具有标志性意义。由于该系统具有许多弱点，引发了后续的许多改进的逻辑系统。人们把这种类似于 BAN 逻辑，用于认证协议安全性证明的逻辑，通称为 BAN 类逻辑。

在所有 BAN 类逻辑中，SVO 是少有的几个具有模型论语义的逻辑，并证明了它的可靠性。这也是这里选择介绍这个逻辑的缘由。

13.3.1 SVO 逻辑的语法

这个逻辑的语法是由两部分组成。一部分是消息语言，另一部分是公式语言。两者都是通过项集合 上的符号相互递归生成的。

SVO 语言的非逻辑符号如下：

(1) $P,Q,\cdots$表示协议参与者的身份；

(2) 密钥符号为：$k,k_1,k_2,\cdots$；

(3) $*_1,*_2,\cdots$；

(4) 函数符号如下：$(X_1,X_2,\cdots,X_n)$是对于 n 个消息的连接运算；

$\{X\}_k$ 用密钥 k 作用到消息 X 上(加密或解密)；

$[X]_k$ 用密钥 k 对于 X 的签名。

(5) 关系符号：**sees**(P,X)、**received**(P,X)、**says**(P,X)、**said**(P,X)、**fresh**(X)。$P\overset{k}{\longleftrightarrow}Q$、$PK_{\psi}(P,k)$、$PK_{\sigma}(P,k)$和 $PK_{\delta}(P,k)$。

我们经常会把 **sees**(P,X)、**received**(P,X)、**says**(P,X)、**said**(P,X)、**believes**(P,X)和 P **controls** 分别写成 P **sees** X、P **received** X、**P** **says** X、P **said** X、P **believes** X 和 P **controls** X 等。

定义 13.3.1 项集合 由下列常量符号组成：

(1) 表示协议参与者身份的常量 $P,Q,\cdots$；

(2) 表示共享密钥、公开密钥和私密密钥的常量符号 $k,k_1,k_2,\cdots$；

(3) 表示数字的常量符号 $1,2,3,\cdots$；

(4) 表示无法识别的符号 $*_1,*_2,\cdots$。

定义 13.3.2 SVO 的消息语言 $\mathcal{M}$ 及公式语言 $\mathcal{F}$ 通过下列方式相互递归生成。

(1) 消息语言 $\mathcal{M}$ 是由下列方式生成的 上的最小语言：

- 如果 $X \in$ ，则 X 是一个消息；
- 如果 $X_1, X_2, \cdots, X_n$ 是消息，并且 F 是任何函数，那么 $F(X_1, X_2, \cdots X_n)$ 是一个消息；
- 如果 φ 是一个公式，则 φ 是一个消息。

(2) 公式语言 $\mathcal{F}$ 是通过下述方式生成的最小的语言：

- 当 P 和 Q 是主体，k 是一个密钥，则 $P \overset{k}{\longleftrightarrow} Q$，$\mathrm{PK}_{\psi}(P,k)$，$\mathrm{PK}_{\sigma}(P,k)$ 和 $\mathrm{PK}_{\delta}(P,k)$ 都是公式；
- 当 X 和 Y 是消息，且 k 是密钥，则 $\mathrm{SV}(X,k,Y)$ 是一个公式；
- 当 X 是一个消息，P 是一个主体，则 P **sees** X、P **received** X、P **said** X 及 **fresh**(X) 都是公式；
- 如果 φ 和 ψ 是公式，则 $\varphi \wedge \psi$ 和 $\neg \varphi$ 也是公式（其他连接词同样定义）；
- 如果 φ 是公式，则 P **believes** φ 和 P **controls** φ 也是公式。

对于上述的基本公式给予直观的解释：

(1) P **controls** φ 表明 P 对于 φ 是可信权威。如果有 P **says** φ，那么 φ 就是被 P 所宣称为正确的。

(2) $P \overset{k}{\longleftrightarrow} Q$ 表明密钥 k 是 P 和 Q 分享的对称密钥。仅有 P、Q 及其所信任的人能够使用这个密钥进行加、解密。

(3) $\mathrm{PK}(P,k)$ 表明 k 是 P 的公钥，密钥 k^{-1} 是对应的私钥。$\mathrm{PK}_{\psi}(P,k)$、$\mathrm{PK}_{\sigma}(P,k)$ 和 $\mathrm{PK}_{\delta}(P,k)$ 分别表示 k 分别是 P 的加密、签名以及密钥协商密钥。

(4) $\mathrm{SV}(X,k,Y)$ 是描述签名验证的。X 是消息 Y 的签名，k 是验证密钥。

(5) 消息 $[X]_k$ 被用以表明使用密钥 k 对于消息 X 的加密及消息 X 本身。这里要说明的是，不使用加、解密函数 $\{X\}_k$ 表示签名，是因为许多的签名方案并不能够通过验证密钥恢复出原文。因而不一定是加、解密运算。

13.3.2　SVO 逻辑推演法则和公理

SVO 逻辑作为一个模态逻辑，具有两个推演法则：

假言推理(MP)：$\dfrac{\varphi, \varphi \rightarrow \psi}{\psi}$。

必然法则(Nec.)：如果 φ 则 P **believes** φ。

注意这里的 φ 的意思是 φ 可以从公理中推演得到。SVO 的公理包括所有命题逻辑的重言式以及下述 20 个公理概型。

信念：对于主体 P 及公式 φ

公理 1. P **believes** $\varphi \wedge P$ **believes**$(\varphi \rightarrow \psi) \rightarrow P$ **believes** ψ。

公理 2. P **believes** $\varphi \rightarrow P$ **believes**$(P$ **believes** $\varphi)$。

公理 1 表明，一个主体相信自己的信念导致的结论。而公理 2 是说每个主体可以分辨自己的信念。

消息源：密钥用于推导消息发送者的身份。

公理 3. $(P \overset{k}{\longleftrightarrow} Q \wedge P$ **received** $\{X^Q\}_k) \rightarrow (Q$ **said** $X \wedge Q$ **sees** $k)$

公理 4. $(\mathrm{PK}_\sigma(Q,k) \wedge R$ **received** $X \wedge \mathrm{SV}(X,k,Y)) \rightarrow Q$ **said** Y

公理 3 表明，如果 P 和 Q 共享有一个好的密钥 k，并且 P 收到来自 Q 的用 k 对 X 加密的密文，则可得出 Q 说了 X 及看见了 k。

公理 4 表明，密钥 k 是 Q 的签名密钥，R 收到了 X，并且通过使用 k 验证说明 X 是 Y 的签名，则得出 Q 说过 Y 的结论。

密钥协商：由好的密钥协商密钥得到的是好的会话密钥。

公理 5. $(\mathrm{PK}_\delta(P,k_p) \wedge \mathrm{PK}_\delta(Q,k_q)) \rightarrow P \overset{F_0(k_p,k_q)}{\longleftrightarrow} Q$

公理 6. $\varphi \Leftrightarrow \varphi[F_0(k,k')/F_0(k',k)]$

其中的 $F_0(k,k')$ 是一个密钥协商函数。比如，Diffie-Hellman 密钥交换协议中的指数函数。公理 6 表明了密钥协商的对称性。

收到的消息：一个主体收到消息的连接、可以解密的消息的密文及签名的消息，则收到该消息。

公理 7. P **received** $(X_1,X_2,\cdots,X_n) \rightarrow P$ **received** X_i

公理 8. $(P$ **received** $\{X\}_k \wedge P$ **sees** $\widetilde{k}) \rightarrow P$ **received** X

公理 9. P **received** $[X]_k \rightarrow P$ **received** X

在公理 8 以及以后文中，$\widetilde{k}$ 表示密钥 k 的逆。如果 k 是公钥，则 $\widetilde{k}$ 表示 k^{-1}，如果 k 是对称密钥，那么 $\widetilde{k}$ 就是 k 本身。

所见到的消息：一个主体见到所有收到的消息，以及从收到的消息能够计算得到的消息。见到消息和收到消息的区别在后面要讲到。

公理 10. P **received** $X \rightarrow P$ **sees** X

公理 11. P **sees** $(X_1,X_2,\cdots,X_n) \rightarrow P$ **sees** X_i

公理 12. P **sees** $X_1 \wedge P$ **sees** $X_2 \wedge \cdots \wedge P$ **sees** $X_n \rightarrow P$ **sees** $F(X_1,X_2,\cdots,X_n)$

这里的 F 是主体 P 可计算的函数。

领会消息：如果一个主体领会一个消息，并看见它的一个函数形式，那么就理解他的确看见的是该消息。

公理 13. P **believes** $(P$ **sees** $F(X)) \rightarrow P$ **believes** $(P$ **sees** $X)$

这里的 F 是 P 能够计算 F 或 F^{-1} 的一个函数。比如，F 可能是一个加密、解密函数。这个公理初看起来说明 P 可以计算 F 的逆：给了 $F(X)$ 可以得到 X。但是，在 P 无法计算 F 的逆时，这个公理说明在 P 具有 X 的情况下，仅仅知道他具有（认识）消息 $F(X)$（这是可识别性）。公理 13 的逆是一个定理。

说明消息：一个主体说过一个连接的消息，那么他就说过所连接的每个消息。最近所说的消息，曾经也说过。任何人都见到自己所说的消息。

公理 14. P **said** $(X_1,X_2,\cdots,X_n) \rightarrow (P$ **said** $X_i \wedge P$ **says** $X_i)$

公理 15. P **says** $(X_1,X_2,\cdots,X_n) \rightarrow P$ **said** $(X_1,X_2,\cdots,X_n) \wedge P$ **says** X_i

权属：这个公理实际是说，P 的话对于问题中的 φ 是法律。

公理 16. (P **control** $\varphi \wedge P$ **says** $\varphi) \rightarrow \varphi$

新鲜性：如果其中有一个消息是新鲜的，那么连接的消息是新鲜的。新鲜消息的有效的 1—1 函数(包括加密和解密函数)值是新鲜的。

公理 17. **fresh**$(X_i) \rightarrow$ **fresh**$(X_1, X_2, \cdots, X_n)$

公理 18. **fresh**$(X_1, X_2, \cdots, X_n) \rightarrow$ **fresh**$(F(X_1, X_2, \cdots, X_n))$

这里的函数值 $F(X_1, X_2, \cdots, X_n)$必须是真正依赖于新鲜消息的，无法保证新鲜性。例如，如果 X 是新鲜的，而 Y 不是新鲜的，那么 $0 \times X + Y$ 就不是新鲜的。

时鲜值的验证：新鲜性将一个消息从过去曾经说过，提升到现在说过。

公理 19. (**fresh**$(X) \wedge P$ **said** $X) \rightarrow P$ **says** X

共享密钥良好性的对称：

公理 20. $P \stackrel{k}{\longleftrightarrow} Q \Leftrightarrow Q \stackrel{k}{\longleftrightarrow} P$

13.3.3 SVO 逻辑的语义

本节简单地介绍 SVO 逻辑语义。SVO 逻辑是在所有安全协议验证的逻辑中为数不多的、具有严格语义的逻辑。由于这个语义定义，以及可靠性的证明，使得 SVO 逻辑的推理具有可信性。有鉴于本章主要目的在于 SVO 逻辑的应用，限于篇幅，在这里将舍去语义定义中较为复杂的部分，以及可靠性证明部分。重点详细介绍应用 SVO 逻辑证明协议安全性质的原理和方法。

1. 计算模型

假设协议的计算是在 n 个主体 $P_1, P_2, \cdots, P_n$ 之间进行的，此外有一个附加的主体 P_e 是环境主体。环境主体可以描述入侵者以及正在传输过程中的消息。

每个主体 P_i 有一个局部状态 s_i。这 $n+1$ 个局部状态构成计算的全局状态。每个主体在计算中体现为 3 种动作：

- 发送一个消息到集合 G 中的主体，记为 send(X, G)；
- 接收某个(些)消息，记为 receive(X)；
- 生成一个消息，记为 generate(X)。

需要注意的是，主体的内部操作，如加密、解密、生成随机数等动作不在协议运行中体现出来，而是通过生成消息而蕴含地体现出来。而一个主体能够生成的消息只能够是原始项，即 中元素。

协议的一次运行是全局状态的一个无限序列。这个序列是通过整数作为时间的标记而记录的。初始状态的时间为 $t=0$，对一个运行 r，在时间 t 的全局状态为 $r(t)$，相对应于它的 P_i 局部状态记为 $r_i(t)$，有时将 $r_i(t)$记为 $r(t)$。

主体可能的变换：对于任意主体 P_i，可能进行的变换就是任意多次使用消息语言定义中的规则，对于目前所接收到的消息进行运算，并且每个主体有能力认识经过不同途径形成的消息的异同。

局部状态：主体 P_i 的局部状态由两部分组成，一部分是主体到目前所进行的所有动作；另一部分是到目前为止所有变换。

环境主体的局部状态由三部分组成：全局动作的历史和环境状态可能的变换，以及对于每个主体 P_i 一个缓冲器 m_i，记录发送给 P_i 但是还没有收到的消息。规定所有的消息，先发后至。即如果 receive(X) 在某个运行 $r_i(t)$ 中出现，则一定存在一个运行 $r_j(t')$，使得 send(X,G) 出现，并且 $t'<t$。

直接收到的消息：一个主体 P_i 除了对于收到的消息进行变换得到新的消息外，主要通过接收外部消息。这些消息包括：

(1) 所有 receive(X) 出现于局部消息中的 X；

(2) 所收到的消息的连接；

(3) 收到 $\{X\}_k$，并且可以运用 $\tilde{k}$ 对于它进行变换得到 X；

(4) 收到某个密钥 k 的签名 $[X]_k$。

所见到的消息：在某个时间点，主体 P_i 直接收到的消息、新生成的消息，以及协议的最初所具有的消息构成 P_i 在这一点所见到的消息。

间接收到的消息：间接收到的消息是没有直接收到，但是属于见到的消息范畴的消息。所收到的消息则是直接和间接收到的消息的总称。

曾声明的消息：曾声明的消息是所见到消息的子集合。给定 P_i 在 (r,t) 所发送的消息 M，定义曾声明的 M 的子消息如下：

(1) M 的所有子消息连接的连接；

(2) M 的没有解密的子消息，并且 P_i 具有它的密钥，而且见到过它的明文；

(3) M 的一个签名子消息的未签名时消息，P_i 有签名密钥，见过该消息；

(4) M 的一个 Hash 消息原像，P_i 见过该原像；

(5) M 的任何子消息 M'，P_i 通过 M 的子消息来说明它。

P_i 在 (r,t) 曾声明过的消息的集合就是在 r 中到 t 为止所发送的所有消息的曾声明的子消息的并。

2. 真值的定义

本节定义一个公式的真值。为此确定一个系统为一些运行的集合 $\mathcal{R}$。如果公式 φ 在某点 (r,t) 真，记为 $(r,t)\ \ \varphi$。与前面一样，φ 表明 φ 是在任何运行下的任何一点都真。

逻辑连词：

$(r,t)\ \ \varphi\wedge\psi$ 当且仅当 $(r,t)\ \ \varphi$ 且 $(r,t)\ \ \psi$。

$(r,t)\ \ \neg\varphi$ 当且仅当 $(r,t)\ \ \varphi$。

接收：$(r,t)\ \ P$ **received** X

当且仅当 X 是 P 在 (r,t) 点接收到的消息集合中元素。

看见：$(r,t)\ \ P$ **sees** X

当且仅当 X 位于 P 在 (r,t) 点所见到的消息集合中。

声明：$(r,t)\ \ P$ **said** X

当且仅当在运行 r 中，某个时刻 $t'<t$ 主体 P 发送消息 M，而 X 是 M 在 (r,t') 点 P 的声明子消息。

$(r,t)\ \ P$ **says** X

当且仅当在 r 的某点 $0\leqslant t'\leqslant t$，P 发送了 M，而 X 是 P 在 (r,t') 的 M 的声明子消息。

权属：(r,t) P **controls** φ

当且仅当 (r,t) P **says** φ 蕴含着对于所有 $0\leqslant t'$，(r,t') P **says** φ。

新鲜性：(r,t) **fresh**(X)

当且仅当对于所有的 P 以及任何的时刻 $t'<0$，都有 (r,t') P **said** X。

密钥：

(1) (r,t) $P\xleftrightarrow{k}Q$

当且仅当对任意时刻 t'，(r,t') R **said**$\{X^Q\}_k$ 意味着 (r,t') R **received**$\{X^Q\}_k$，或者 $R=Q$ 且 (r,t') R **said** X 且 (r,t') R **sees** k。

如果 (r,t') R **received**$\{X\}_k$，则 $R\in\{P,Q\}$。

(2) (r,t) $\mathrm{SV}(Y,k,X)$ 当且仅当存在密钥 $\tilde{k}$ 可以使用 K 验证 $Y=[X]_{\tilde{k}}$。

(3) (r,t) $\mathrm{PK}_{\sigma}(P,K)$ 当且仅当对于所有 t'，(r,t') Q **received** $Y\wedge\mathrm{SV}(Y,k,X)\Rightarrow(r,t')$ P **said** X。

(4) (r,t) $\mathrm{PK}_{\psi}(P,K)$ 当且仅当对于所有 t'，(r,t') Q **sees**$\{X\}_k\Rightarrow$ 仅当 $Q=P$ 时，(r,t') Q **sees** X。

(5) (r,t) $\mathrm{PK}_{\delta}(P,K)$ 当且仅当对于所有的 t'：

① 对某个 Q 和 k_q，(r,t') $P\xleftrightarrow{F_0(k,k_q)}Q$；

② 对于所有的 R、k_r，如果 (r,t') $R\xleftrightarrow{F_0(k,k_r)}P$，则对于任意 U、k_u，(r,t') $R\xleftrightarrow{F_0(k_u,k_r)}U$。

其中的 F_0 函数是协议中的密钥协商函数。

信念：(r,t) P_i believes φ 当且仅当对于所有满足 $(r,t)\sim_i(r',t')$ 的 (r',t') 有 (r,t) $\varphi_i(r',t')$，并且对于某个这样的 (r',t')，有 $\varphi=\varphi_i(r',t')$。

限于篇幅，对于信念的定义这里省略了一些内容。请读者参考文献[4]。

根据上述真值的定义，可以证明以下定理。

定理 13.3.1 SVO 是可靠的：如果 Γ φ，则 Γ φ。

这个定理表明，协议验证的过程中所进行的推演都是可信的。从而保证不会推演出荒谬的结论。这里同样省略了定理的证明过程。

13.4 利用 SVO 逻辑分析协议的原理

利用 SVO 逻辑对于协议的分析与 BAN 逻辑最大的区别在于：SVO 逻辑无须对于协议进行理想化。这是一个较大的改进，因为理想化的过程是一个困难的、容易出现错误的人为过程。

对于安全协议的语法分析主要由两步构成。

(1) 前提建立：这些前提体现协议描述的假设。

(2) 目标证明：利用这些前提、公理及逻辑推演法则证明所要达到的目标。

这两步的目标，就是看是否能够从前提到达协议要达到的目标。如果无法证明协议的安全目标，就要考虑任何增加一些步骤才能够达到这个或这些目标。这个过程就包含着寻找协议攻击的可能性。

在这两步中，建立前提的过程是一个相对较为复杂的过程。为了便于读者理解，以用 Needham-Schröder 密钥分发协议为例，说明第一步建立前提的过程。下一节将会完整的分析一个密钥协商协议。

Needham-Schröder 是协议分析的一个标准协议。协议本身含有以下 5 步：

(1) $A \rightsquigarrow S$：A, B, N_a；

(2) $S \rightsquigarrow A$：$\{N_a, B, k_{ab}, \{k_{ab}, A\}_{k_{bs}}\}_{k_{as}}$；

(3) $A \rightsquigarrow B$：$\{k_{ab}, A\}_{k_{bs}}$；

(4) $B \rightsquigarrow A$：$\{N_b\}_{k_{ab}}$；

(5) $A \rightsquigarrow B$：$\{N_b - 1\}_{k_{ab}}$。

符号解释：A、B 是两个主体的身份标识，N_a、N_b 分别是它们的时鲜值，k_{ab} 是服务器 S 分发给他们的共享的密钥。k_{as} 和 k_{bs} 分别是 A 和 B 与 S 共享的密钥。$\{X\}_k$ 表示用密钥 k 对于信息 X 加密的密文。

前提一般分为以下 4 类。

第一类前提是初始假设，就是协议开始时假设正确的假设。包括主体相信自己生成的时鲜值的新鲜性，自己与服务器所共享的密钥的良好性。服务器对于自己生成的会话密钥的权属及良好性。还包括主体对于所具有的项的领会，以及对于签名验证的领会。

对于 Needhan-Schröder 协议，有下面的假设前提：

P1　A **believes fresh**(N_a)

　　B **believes fresh**(N_b)

P2　A **believes** S **control**$(A \xleftrightarrow{k_{ab}} B)$

　　B **believes** S **control**$(A \xleftrightarrow{k_{ab}} B)$

P3　A **believes** S **control**(**fresh**(k_{ab}))

　　B **believes** S **control**(**fresh**(k_{ab}))

P4　A **believes**$(A \xleftrightarrow{k_{as}} S)$

　　B **believes**$(B \xleftrightarrow{k_{bs}} S)$

第二类前提是关于协议运行时所收到信息的前提。这类假设课题直接从协议的描述中得到，并且在证明中很少用到。但是可以用于形成其他后续的假设。

在 Needham-Schröder 协议中，这类的假设是：

P5　A **received**$\{N_a, B, k_{ab}, \{k_{ab}, A\}_{k_{bs}}\}_{k_{as}}$

P6　B **received**$\{k_{ab}, A\}_{k_{bs}}$

P7　A **received**$\{N_b\}_{k_{ab}}$

P8　B **received**$\{N_b - 1\}_{k_{ab}}$

第三类前提是关于主体对于所收到信息的领会或理解。即使一个主体收到一个消息，也无法保证他对于这个消息完全了解。比如所分发的密钥，由于其随机性，使得主体难以识别它。这类的前提可以容易地从上一类前提中得到。

具体到 Needham-Schröder 协议有

P9　A **believes** A **received** $\{N_a, B, *_1, *_2\}_{k_{as}}$

P10　B **believes** B **received** $\{*_3, A\}_{k_{bs}}$

P11　B **believes** B **received** $\{N_b - 1\}_{*_3}$

注意这里没有对于对应于 P7 的前提假设，这是因为 A 没有从该消息中理解任何东西。公式中的“$*$”表示的是无法识别的消息内容。

第四类前提是每个主体对于收到的消息进行的解释。就是主体认为发送者发送这个消息的含义。

具体到 Needham-Schröder 协议有

P12　A **believes**(A **received** $\{N_a, B, *_1, *_2\}_{k_{as}} \rightarrow$

A **received** $\{N_a, B, A \xleftrightarrow{k_{ab}} B, \textbf{fresh}(k_{ab}), *_2\}_{k_{as}}$)

P13　B **believes**(B **received** $\{*_3, A\}_{k_{bs}} \rightarrow B$ **received** $\{A \xleftrightarrow{k_{ab}} B, \textbf{fresh}(k_{ab})\}_{k_{bs}}$)

P14　B **believes**(B **received** $\{*_3, A\}_{k_{bs}} \wedge B$ **received** $\{N_b - 1\}_{*_3} \rightarrow$

B **received** $\{N_b - 1\}_{k_{bs}}$)

注意在 P14 中，B 要相信自己收到 $\{N_b - 1\}_{k_{bs}}$，首先必须收到该消息，而且要收到 S 告诉他有关 k_{ab} 的消息。没有后者，就无法识别前者。

一旦这些前提设定以后，就可以应用逻辑公理和推演法则进行协议目标的推演。这里要注意的是，SVO 逻辑除了假言判断以外含有一个必然法则。这个必然法则必须只能够对于逻辑定理来引用，不能够对于一般的结论使用；否则就会推演出荒谬的结论。而逻辑定理就是仅仅由逻辑公理和推演法则得到的那些句子，即一个公式 φ 是一个定理当且仅当 φ。

13.5　一个密钥协商协议的逻辑分析过程

密钥协商协议的分析要较之认证协议的分析更为微妙。主要原因在于密钥协商需要对于更多的安全性质，诸如密钥的确认、密钥良好性以及长期、短期密钥的绑定等性质的描述。

13.5.1　协议分析常用的协议目标

下面是协议分析时常用的协议目标的描述。

G1．远端的操作性：A **believes** B **says** X

G2．身份认证：A **believes** B **says** $F(X, N_a)$

G3．安全密钥建立：A **believes** $\left(A \xleftrightarrow{k} B \wedge A \textbf{ sees } k\right)$

G4．密钥确认：A **believes** $\left(A \xleftrightarrow{k} B \wedge A \textbf{ sees } k \wedge U \textbf{ says}(U \textbf{ sees } k)\right)$

G5．密钥的新鲜性：A **believes fresh**(k)

G6．共享密钥的相互领会：A **believes** B **says**$\left(A \overset{k}{\longleftrightarrow} B \wedge A \text{ sees } k\right)$

这些是一些主要的安全目标，但并非所有密钥协商协议都能够或者必须达到所有的目标。大部分的目标从表达上就可以清楚其意义。G1 说明 A 相信目前 B 是在线的。在 G2 中 N_a 是 A 的时鲜值，函数 F 是 B 可以有效计算的 1－1 函数，并且 A 可以计算 F 或 F 的逆。基本思想是，A 确信 B 最近对于 A 的挑战时鲜值 N_a 有一个回应 X，而 G3 在 B 没有参与目前协议甚至没有密钥 k 的情况下也可能成立。

下面是一些典型的协议初始假设。这些假设涉及可信的权威 T 和协议参与者 A。在具体协议中它们可能是关于所有参与者都成立的。

A1．T 的签名密钥：A **believes** $\mathrm{PK}_\sigma(T, k_t)$

A2．T 的签名密钥的所属：A **believes** T **control** $\mathrm{PK}_\sigma(B, k_b)$

A3．T 的协商密钥的所属：A **believes** T **control** $\mathrm{PK}_\delta(B, k_b)$

A4．自己协商密钥的良好性：A **believes** $\mathrm{PK}_\delta(A, k_a)$

A5．时鲜值的新鲜性：A **believes fresh**(N_a)

这些假设的意义明显：主体相信具有可信权威的良好签名密钥；可信权威具有其他主体公开密钥论断的所属权。每个主体相信自己密钥协商密钥的良好性，以及自己所生成时鲜值的新鲜性。

有一点要注意的是，所属权是很强的概念，使用时需要倍加小心。可信权威在发放公钥证书时，不仅要确认所请求的公钥，而且要确认请求者具有相应的私钥。对于签名密钥以及密钥协商密钥来说尤为重要。

13.5.2 MTI 协议的描述

下面用 SVO 逻辑分析 MTI 密钥协商协议。MTI 密钥协商协议是由 Matsumoto、Takashima 和 Imai 在 1986 年提出的。协议的目的是建立一个共享密钥：使用两个 Diffie-Hellman 指数运算，结合事先确定的变元，最终生成一个共享密钥。

首先，协议的主体都具有可信方 T 发放的公钥证书。他们共享一个密码系统的参数：一个 p 阶群 G，其中离散对数问题是难解的，以及 G 的一个生成元 g。主体 A 的私钥为 x，B 的私钥为 y。他们的公钥分别为 $X=g^x$ 和 $Y=g^y$。记主体 A 的公钥证书为 $\mathrm{Cert}_a=\{A, X, [X, A]_{k_t^{-1}}\}$，主体 B 的公钥证书为 $\mathrm{Cert}_a=\{B, Y, [B, Y]_{k_t^{-1}}\}$。协议的信息发送只有一个来回：

$$A \leadsto B: \mathrm{Cert}_a, R_a$$
$$B \leadsto A: \mathrm{Cert}_b, R_b$$

其中 A 选择一个任意数 r，计算 $R_a=g^r$；主体 B 选择一个任意数 s，计算 $R_b=g^s$。双方各向对方发送自己的公钥证书及所计算的 R 值。

在验明对方的公钥证书后，主体 A 计算 $k=Y^r R_b^x=g^{sx+ry}$，主体 B 计算 $k=X^s R_a^y=g^{sx+ry}$。这个公共的值就是共享密钥。用形式符号表示上述协议就是：

$$A \rightsquigarrow B:(A,X,[A,X]_{k_t^{-1}}),R_a$$
$$B \rightsquigarrow A:(B,Y,[B,Y]_{k_t^{-1}}),R_b$$

1. 协议前提的建立

为了证明协议的目标，首先建立协议的假设。假设每个主体相信 k_t 是可信权威 T 的签名验证密钥(A1)：

P1. A **believes** $\mathrm{PK}_\sigma(T,k_t)$

B **believes** $\mathrm{PK}_\sigma(T,k_t)$

每个主体相信个人密钥的良好性(A4)：

P2. A **believes** $\mathrm{PK}_\delta(A,(X,R_a))$

B **believes** $\mathrm{PK}_\delta(B,(Y,R_b))$

每个主体相信自己生成的随机值是新鲜的（A5)：

P3. A **believes fresh**(R_a)

A **believes fresh**(R_b)

我们要表达：每个主体相信，可信权威对于另一方的协商密钥有所属权。但是，在本协议中，A3 是不适合的。主要原因在于，所属的概念是通过控制来表述的。而控制的语义是，可信权威对于所生成的信息具有全部的控制权，并且是新鲜的。本协议中的协商密钥中公钥证书没有新鲜性保证，并且协商密钥不仅仅依赖于公钥，而且依赖于主体生成的新的随机值。然而可以用下述的

P4. A **believes**$((T$ **said** $\mathrm{PK}_\delta(B,Y))\wedge$

A **received**$((B,Y,[B,Y]_{k_t^{-1}}),*_b)\rightarrow\mathrm{PK}_\delta(B,(Y,*_b)))$

B **believes**$((T$ **said** $\mathrm{PK}_\delta(A,X))\wedge$

B **received**$((A,X,[A,X]_{k_t^{-1}}),*_a)\rightarrow\mathrm{PK}_\delta(A,(X,*_a)))$

以上的前提是协议初始假设形成的前提。如 13.4 节所述，还有存在反映主体收到消息的前提、对于所收到消息的理解的前提，以及主体对于收到的消息的解释形成的前提。把它们一起罗列如下：

P5. A **believes** A **sees** (X,R_a,x,r)

B **believes** B **sees** (Y,R_b,y,s)

P6. A **believes** $\mathrm{SV}([B,Y]_{k_t^{-1}},k_t,(B,Y))$

B **believes** $\mathrm{SV}([A,X]_{k_t^{-1}},k_t,(A,X))$

P7. A **received**$((B,Y,[B,Y]_{k_t^{-1}}),R_b)$

B **received**$((A,X,[A,X]_{k_t^{-1}}),R_a)$

P8. A **believes** A **received** $((B,Y,[B,Y]_{k_t^{-1}}),*_b)$

B **believes** B **received** $(A,X,[A,X]_{k_t^{-1}}),*_a)$

P9. A **believes**$(T$ **said**$(B,Y)\rightarrow(T$ **said** $\mathrm{PK}_\delta(B,Y)))$

B **believes**$(T$ **said**$(A,X)\rightarrow(T$ **said** $\mathrm{PK}_\delta(A,X)))$

下面的任务就是推演证明协议的目标。由于本协议没有认证的内容，无法推演出上面给出的目标，即 G1、G2、G4 及 G6。只能证明协议的目标 G3 和 G5。

2. 协议安全目标的证明

根据上述前提，证明下面的协议目标：

G3. 安全密钥建立：A **believes** $\left(A \xleftrightarrow{k} B \wedge A \textbf{ sees } k\right)$

B **believes** $\left(A \xleftrightarrow{k} B \wedge A \textbf{ sees } k\right)$

其中 $k=F_0((X,R_a),(Y,R_b))$，这里的 F_0 是协议中的指数的乘法函数。

G5. 密钥的新鲜性：A **believes fresh**(k)

B **believes fresh**(k)

由于协议对于 A 和 B 是对称的，只需对于 A 进行推演证明即可。则记上述的所有前提的集合为 Γ。

G3. 的推演(见表 13.11)：$\Gamma\ \ A$ **believes** $\left(A \xleftrightarrow{k} B \wedge A \textbf{ sees } k\right)$

表 13.11 G3 的推演

推演过程	论据
记 $\varphi_1 = A$ **received**$((B,Y,[B,Y]_{k_t^{-1}}), *_b)$， $\psi_1 = A$ **received** $[B,Y]_{k_t^{-1}}$	
1. $\Gamma\ \ A$ **believes** φ_1	假设 P8
2. $\Gamma\ \ \varphi_1 \rightarrow \psi_1$	公理 7
3. $\Gamma\ \ A$ **believes**$(\varphi_1 \rightarrow \psi_1)$	Nec. 法则
4. $\Gamma\ \ A$ **believes** $\varphi_1 \wedge A$ **believes**$(\varphi_1 \rightarrow \psi_1) \rightarrow A$ **believes** ψ_1	公理 1
5. $\Gamma\ \ A$ **believes** A **received**$[B,Y]_{k_t^{-1}}$	1,3,对 4 用 MP,代入 ψ_1
6. $\Gamma\ \ A$ **believes** $PK_\sigma(T,k_t)$	假设 P1
7. $\Gamma\ \ A$ **believes** $SV([B,Y]_{k_t^{-1}}, k_t, (B,Y))$ 令 $\varphi_2 = (A$ **received** $[B,Y]_{k_t^{-1}}) \wedge PK_\sigma(T,k_t) \wedge SV([B,Y]_{k_t^{-1}}, k_t, (B,Y))$	假设 P6
8. $\Gamma\ \ A$ **believes** φ_2	5,6,7
9. $\Gamma\ \ \varphi_2 \rightarrow T$ **said**(B,Y)	公理 4
10. $\Gamma\ \ A$ **believes**$(\varphi_2 \rightarrow T$ **said**$(B,Y))$	Nec.
11. $\Gamma\ \ (A$ **believes** $\varphi_2 \wedge A$ **believes**$(\varphi_2 \rightarrow T$ **said**$(B,Y)))$ $\rightarrow A$ **believes** $(T$ **said** (B,Y)	公理 1
12. $\Gamma\ \ A$ **believes**$(T$ **said**$(B,Y))$	8,10 对 11 用 MP
13. $\Gamma\ \ A$ **believes**$(T$ **said**$(B,Y) \rightarrow (T$ **said** $PK_\delta(B,Y)))$	假设,P9
14. $\Gamma\ \ A$ **believes** $(T$ **said** $PK_\delta(B,Y))$	12,13 对公理 1 用 MP
15. $\Gamma\ \ A$ **believes**$((T$ **said** $PK_\delta(B,Y)) \wedge$ A **received**$((B,Y,[B,Y]_{k_t^{-1}}), *_b) \rightarrow PK_\delta(B,(Y, *_b)))$	假设,P4
16. $\Gamma\ \ A$ **believes** $PK_\delta(B,(Y, *_b))$	14,15,P8 对公理 1 用 MP

续表

推演过程	论　据
17. Γ A **believes** $PK_\delta(A,(X,R_a))$	假设,P2
18. Γ $(PK_\delta(B,(Y,*_b))\wedge PK_\delta(A,(X,R_a))\rightarrow A\xleftrightarrow{k}B$ 其中 $k=F_0((X,R_a),(Y,R_b))$	公理 5
19. Γ A **believes**$((PK_\delta(B,(Y,*_b))\wedge PK_\delta(A,(X,R_a))\rightarrow A\xleftrightarrow{k}B)$	Nec.
20. Γ A **believes** $A\xleftrightarrow{k}B$	16,17,19 对公理 1 用 MP
21. Γ A **believes** A **received**$((B,Y,[B,Y]_{k_t^{-1}}),*_b)$	假设 P8
22. Γ A **believes** $(A$ **received**$((B,Y,[B,Y]_{k_t^{-1}}),*_b)\rightarrow$ A **sees**$((B,Y,[B,Y]_{k_t^{-1}}),*_b)$	公理 11 和 Nec.
23. Γ A **believes** A **sees**$((B,Y,[B,Y]_{k_t^{-1}}),*_b)$	21,22 对公理 1 用 MP
24. Γ A **believes**$(A$ **sees**$((B,Y,[B,Y]_{k_t^{-1}}),*_b)\rightarrow A$ **sees** $Y)$	公理 11 和 Nec.
25. Γ A **believes** A **sees** Y	23,24 对公理 1 用 MP
26. Γ A **believes**$(A$ **sees**$((B,Y,[B,Y]_{k_t^{-1}}),*_b)\rightarrow A$ **sees** $*_b$	公理 11 和 Nec.
27. Γ A **believes** A **sees**, $*_b$	23,26 对公理 1 用 MP
28. Γ A **believes**$(A$ **sees** $Y\wedge A$ **sees** $*_b\rightarrow A$ **sees**$(Y,*_b))$	公理 12 和 Nec.
29. Γ A **believes** A **sees**$(Y,*_b)$	25,27 对公理 1 用 MP
30. Γ A **believes** A **sees**(X,R_a,X,r)	假设 P5
31. Γ A **believes**$(A$ **sees** $X\wedge A$ **sees** $R_a\wedge A$ **sees** $x\wedge A$ **sees** $r\wedge A$ **sees** Y $\wedge A$ **sees** $*_b)\rightarrow A$ **sees** k	公理 12 和 Nec.
32. Γ A **believes** A **sees** k 其中 $k=F_0((X,R_a),(Y,R_b))$	29, 30, 31 对公理 1 用 MP
33. Γ A **believes**$(A\xleftrightarrow{k}B\wedge A$ **sees** $k)$	20,32 对公理 1 用 MP

G5. 密钥的新鲜性(见表 13.12): A **believes fresh**(k)的推演。

表 13.12　G5 的推演

推演过程	论　据
1. Γ A **believes fresh**(R_a)	假设 $P3$
2. Γ **fresh**$(R_a)\rightarrow$**fresh**$(F_0((X,R_a),(Y,R_b)))$	公理 18
3. Γ A **believes**(**fresh**$(R_a)\rightarrow$**fresh**$(F_0((X,R_a),(Y,R_b))))$	对 2 用 Nec.
4. Γ A **believes fresh**(k) 其中 $k=F_0((X,R_a),(Y,R_b))$	2,3 对公理 1 用 MP

通过上述两个例子的推演,可以看到逻辑推演并非是一个显然和轻松的事情。往往为了推演一个目标公式,需要许多的中间结果。而这往往是困难的。在逻辑中

还有一种消解法，对于证明系统有极大的方便。读者可以参考后面的参考文献。

13.6 注记

本章主要介绍数理逻辑的基础知识，通过它们在安全协议中的应用，使得读者体会到逻辑系统在安全协议验证中的应用原理。正如本章开始所述，在计算机科学中，定义了形形色色的逻辑系统，以适应各种不同应用领域的需要。在近年来的热点研究中，模型检测的方法得到了长足的发展，其中逻辑系统发挥着不可替代的作用（参见文献[5]）。我们介绍了模型检测的基本原理，但是限于篇幅，没有介绍它们在安全协议验证方面的应用。逻辑系统在访问控制、入侵检测、信息流控制等方面也有重要的应用。这些应用都是在基本逻辑系统的基础上，扩充需要的功能而得到的。所以本章的内容是基础性的结果，是必须掌握的基础知识。

本章的内容主要取材于参考文献[1]、[4]。由于篇幅的限制，在本章简略地介绍了数理逻辑的基本知识。数理逻辑是一个根深叶茂，并且仍然非常活跃的学科，在数学和计算机科学的许多场合中有着重要应用。同时，数理逻辑本身的发展，产生了许多独立于传统数学的工具和方法。希望具备本章的知识，对读者进一步阅读相关文献（如文献[1]）能够起到事半功倍的作用。

参考文献

[1] Shawn Hedman. A First Course in Logic——An Introduction to Model Theory, Proof Theory, Computability, and Complexity. Oxford University Press, 2004

[2] Burrows M, Abadi M, Needham R M. A Logic of Authentication, ACM Transactions on Computer Systems, Vol. 8, No. 1, Feb 1990, pp. 18-36

[3] Paul F. Syverson, Paul C. van Oorschot. On unifying some cryptographic protocols. In Proceedings of the IEEE Computer Society Symposium on Research in Security and Privacy, pages 14-28. IEEE CS Press, May 1994

[4] Paul F. Syverson, Paul C. van Oorschot. A unified cryptographic protocol logic. NRL Publication 5540-227, Naval Research Lab, 1996

[5] Huth M, Ryan M . Logic in Computer Science——Modeling and Reasoning about Systems. 2nd ed. Cambridge University Press, 2004

第 14 章　数字信号处理方法与技术

信息隐藏历史悠久，但现代信息隐藏技术仅起源于 20 世纪 90 年代，它指将特定用途的信息隐藏在其他信息载体中，使得它们难以被消除或发现[1,2]。由于人类感知对数字多媒体（主要包括数字图像、音频和视频）的一些成分变化不敏感，并且它们的应用已经大量普及，现代信息隐藏的一个重要特征是载体数据多为多媒体，这使得信息隐藏和数字信号处理[3~5]有了直接的联系，数字信号处理成为信息隐藏的基础性方法与技术。本章的目的就是介绍信息隐藏涉及的信号处理方法与技术。

现代信息隐藏主要包括数字水印（watermarking）[1]和隐写（steganography）[2]两个领域，前者又包含鲁棒（亦称稳健）水印和脆弱水印。鲁棒水印是重要的多媒体版权保护技术之一，它指将与版权有关的信息隐蔽地嵌入数字内容，攻击者难以在载体不遭到显著破坏的情况下消除水印，使得授权者可以通过水印验证实现对版权所有者或内容购买者的认定。脆弱数字水印技术将防伪信息隐藏在数字内容中，但目的却是以后通过检测发现篡改，由于防伪信息和被保护数据融合，方便地支持了电子图文的流动。隐写指利用可公开的信息隐藏保密的信息，通过隐蔽保密的事实获得新的安全性。

14.1　基本概念

本节将给出有关数字信号的基本概念和方法。

14.1.1　时域离散信号与系统

定义 14.1.1　对模拟信号 $x_a(t)$ 每隔 T 时间采样一次，得到时（空）域离散信号

$$x(n) = x_a(t)\,|_{t=nT} = x_a(nT), \quad n \in \mathbf{Z} \tag{14.1}$$

其中，$\mathbf{Z}$ 表示整数集合；$x(n)$ 是一个按 n 有序的数字序列。

计算机只能处理数字信号[3]，它指时间和幅度均离散的信号，其中幅度取接近的量化值。本章一般假设幅度已经过量化或量化效应对信号处理的影响可忽略，因此，如无特殊说明，以下提到的时域离散信号也指数字信号。

信号可以基本地分为随机信号[4]和确定性信号。前者随着时间的推移没有明显的变化规律，不能用明确的数学关系描述，后者则可以。常用的确定性时域离散信号包括单位冲激序列

$$\delta(n) = \begin{cases} 1 & n = 0 \\ 0 & n \neq 0 \end{cases}$$

矩形序列

$$R_N(n) = \begin{cases} 1 & n \in [0, N-1] \\ 0 & n \notin [0, N-1] \end{cases}$$

正弦序列 $x(n)=\sin(\omega n)$ 与复指数序列 $x(n)=e^{(\sigma+j\omega)n}=e^{\sigma n}(\cos(\omega n)+j\sin(\omega n))$ 等。其中，$\delta(n)$ 与 $R_N(n)$ 是习惯性的标记方法；ω 为数字频率，对于被采样的模拟正弦信号 $x_a(t)=\sin(\Omega t)$，由于 $x(n)=x_a(t)|_{t=nT}=\sin(\Omega nT)=\sin(\omega n)$，因此 $\omega=\Omega T$。

数字信号处理常对时域离散信号施加一些基本操作。对 $x(n)$ 与 $x'(n)$ 执行加法得到 $y(n)=x(n)+x'(n)$，其中两个序列在 n 相同处两两相加；设 $u,v\in\mathbf{Z}$，可通过移位、翻转和缩放将 $x(n)$ 变为 $x(n-u)$、$x(-n)$ 和 $x(vn)$；卷积是另一个常用的操作，设 $m\in\mathbf{Z}$，$x(n)$ 与 $x'(n)$ 之间的卷积被定义为

$$y(n)=\sum_{m=-\infty}^{\infty}x(m)x'(n-m)=x(n)*x'(n) \tag{14.2}$$

多数对时域离散信号的操作或处理可以表示为时域离散系统对这类信号的作用，其中，最常用的是线性时不变系统，它使得对信号的处理和分析变得更简单。设 $T[\cdot]$ 表示系统施加的作用，则输入与输出间的关系为 $y(n)=T[x(n)]$，则对线性时不变系统可定义如下：

定义 14.1.2 设 $y_1(n)=T[x_1(n)]$，$y_2(n)=T[x_2(n)]$。若时域离散系统满足

(1) 可加性：

$$y_1(n)+y_2(n)=T[x_1(n)+x_2(n)] \tag{14.3}$$

(2) 比例性：

$$ay_1(n)=T[ax_1(n)] \tag{14.4}$$

则它是线性系统；设 m 为任意整数，若系统满足

(3) 移位不变性：

$$y(n-m)=T[x(n-m)] \tag{14.5}$$

则它是时不变系统。线性时不变(LTI)系统是满足以上两个要求的系统。

例 14.1.1 设 a 与 b 为常数，现考察仿射系统 $y(n)=T[x(n)]=ax(n)+b$ 是否是 LTI 系统。由于

$$y_1(n)=T[x_1(n)]=ax_1(n)+b$$
$$y_2(n)=T[x_2(n)]=ax_2(n)+b$$
$$y(n)=T[x_1(n)+x_2(n)]=ax_1(n)+ax_2(n)+b\neq y_1(n)+y_2(n)$$

因此仿射系统不是线性系统；由于

$$y(n-n_0)=ax(n-n_0)+b=T[x(n-n_0)]$$

因此仿射系统是时不变系统。

LTI 系统对输入的作用可以用系统的单位冲激响应描述，为此这里给出以下定义。

定义 14.1.3 设时域离散系统 $T[\cdot]$ 的输入为单位冲激序列 $\delta(n)$，系统输出的初态为零，则在此条件下的系统输出 $h(n)=T[\delta(n)]$ 为该系统的单位冲激响应。

定理 14.1.1 时域离散 LTI 系统的输出等于输入与该系统的单位冲激响应的卷积。

证明：设该系统 $T[\cdot]$ 的单位冲激响应为 $h(n)=T[\delta(\mathrm{n})]$，输入为 $x(n)$，则它可表示为

$$x(n)=\sum_{m=-\infty}^{\infty}x(m)\delta(n-m)$$

则系统输出为

$$\begin{aligned}y(n)&=T\Big[\sum_{m=-\infty}^{\infty}x(m)\delta(n-m)\Big]\\&=\sum_{m=-\infty}^{\infty}x(m)T[\delta(n-m)]\\&=\sum_{m=-\infty}^{\infty}x(m)h(n-m)=x(n)*h(n)\end{aligned}\tag{14.6}$$

其中，前一步利用了式(14.3)和式(14.4)，后一步利用了式(14.5)。

对式(14.6)描述的系统，若 $h(n)$ 为有限长度，则称为是有限冲激响应的(FIR)，否则，系统是无限冲激响应的(IIR)。FIR 系统是无反馈的，而 IIR 系统是有反馈的，可用有反馈的常系数线性差分方程

$$y(n)=\sum_{m=0}^{M}b_m x(n-m)-\sum_{m=1}^{N}a_k y(n-k)\tag{14.7}$$

或等价的

$$\sum_{m=0}^{N}a_k y(n-k)=\sum_{m=0}^{M}b_m x(n-m)\quad a_0=1\tag{14.8}$$

表示，其中 a_i 和 b_i 均为常数。当 $a_i=0, 1\leqslant i\leqslant K$，式(14.8)也可描述 FIR 系统。

以上提到的信号或系统均有其 2 维的对应形式。例如，设 2 维 LTI 系统的单位冲激响应为 $h(n_1,n_2)$，当输入为 2 维时域离散信号 $x(n_1,n_2)$时，输出为它们的 2 维卷积

$$y(n_1,n_2)=\sum_{k=-\infty}^{\infty}\sum_{k_2=-\infty}^{\infty}h(n_1-k_1,n_2-k_2)u(k_1,k_2)\tag{14.9}$$

类似地，描述 2 维 LTI 系统的差分方程为

$$\begin{aligned}y(n_1,n_2)=&\sum_{m=0}^{M}\sum_{n=0}^{N}b_{mn}x(n_1-m,n_2-n)\\&-\sum_{m=1}^{M}\sum_{n=1}^{N}a_{mn}y(n_1-m,n_2-n)\end{aligned}\tag{14.10}$$

若将系统的处理看作对信号的变换，则 $h(n_1,n_2)$也被称为变换核。当 $h(n_1,n_2)=h(n_1)h(n_2)$，相应的系统或变换被称为是可分离的。

14.1.2　时域离散信号与系统的频域分析

傅里叶变换与 Z 变换是分析时域离散信号与系统的基本手段。

定义 14.1.4　若 $n=\sum_{-\infty}^{\infty}|x(n)|<\infty$，则

$$X(e^{j\omega})=\mathrm{FT}[x(n)]=\sum_{n=-\infty}^{\infty}x(n)e^{-j\omega n}\tag{14.11}$$

为时域离散信号 $x(n)$的傅里叶变换(FT)。

在以上定义中，$X(e^{j\omega})$被称为傅里叶变换系数，它们反映了 $x(n)$在不同频率上的分布情况，是重要的信号特征；$\sum_{n=-\infty}^{\infty}|x(n)|<\infty$保证了傅里叶变换系数的收敛，因此是该变换存在的前提。用 $e^{j\omega n}$ 乘以式(14.11)两侧并在$(-\pi,\pi)$内对 ω 积分，得到逆傅里叶变换(IFT)

$$x(n)=\mathrm{IFT}[X(e^{j\omega})]=\frac{1}{2\pi}\int_{-\pi}^{\pi}X(e^{j\omega})e^{j\omega n}\,d\omega \tag{14.12}$$

定义 14.1.5 对时域离散信号 $x(n)$，若 $\sum_{n=-\infty}^{\infty}|x(n)z^{-n}|<\infty$，则 $x(n)$的 Z 变换(ZT)为

$$X(z)=\mathrm{ZT}[x(n)]=\sum_{n=-\infty}^{\infty}x(n)z^{-n} \tag{14.13}$$

其中 z 为复变量。

Z 变换也需保证级数和收敛，这要级数绝对可和，即 $\sum_{n=-\infty}^{\infty}|x(n)z^{-n}|<\infty$，称该条件得到满足的 Z 变量取值范围为收敛域，一般表示为 $R_{-x}<|z|<R_{+x}$。在复平面上，类似地用前述获得 IFT 的方法可以得到逆 Z 变换为

$$x(n)=\mathrm{IZT}[X(z)]=\frac{1}{2\pi j}\oint_{c}X(z)z^{n-1}\,dz \quad c\in(R_{-x},R_{+x}) \tag{14.14}$$

显然，Z 变换是傅里叶变换的推广，它们之间的关系是 $X(e^{j\omega})=X(z)|_{z=e^{j\omega}}$。因此，它们都可以用于描述线性时不变系统的频率特征。

定义 14.1.6 设线性时不变系统的单位冲激响应为 $h(n)$，称它的傅里叶变换

$$H(e^{j\omega})=\sum_{n=-\infty}^{\infty}h(n)e^{-j\omega n} \tag{14.15}$$

为系统的传输函数，称它的 Z 变换 $H(z)=\sum_{n=-\infty}^{\infty}h(n)z^{-n}$ 为系统的系统函数。

若由 $h(n)$定义的系统的输入为频率为 ω 的复数信号序列 $x(n)=e^{j\omega n}$，则输出为

$$y(n)=\sum_{k=-\infty}^{\infty}h(k)e^{j\omega(n-k)}=e^{j\omega n}\sum_{k=-\infty}^{\infty}h(k)e^{-j\omega k}=e^{j\omega n}H(e^{j\omega})$$

因此，传输函数也称为频率响应。

例 14.1.2 现推导延迟系统 $y(n)=x(n-n_d)$的频率响应。根据以上分析，可以用复数信号序列 $x(n)=e^{j\omega n}$作为系统输入求得频率响应，因此，令延迟系统的输入是 $x(n)=e^{j\omega n}$，则有

$$y(n)=e^{j\omega(n-n_d)}=e^{-j\omega n_d}e^{j\omega n}$$

因此频率响应为 $H(e^{j\omega})=e^{-j\omega n_d}$。

定理 14.1.2 若 $y(n)=x(n)*h(n)$，则 $Y(e^{j\omega})=H(e^{j\omega})X(e^{j\omega})$。

证明：由于 $y(n)=\sum_{m=-\infty}^{\infty}x(m)h(n-m)$，则

$$Y(e^{j\omega})=\mathrm{FT}[y(n)]=\sum_{n=-\infty}^{\infty}\left[\sum_{m=-\infty}^{\infty}x(m)h(n-m)\right]e^{-j\omega n}$$

记 $k=n-m$，则

$$Y(e^{j\omega}) = \sum_{k=-\infty}^{\infty}\sum_{m=-\infty}^{\infty} h(k)x(m)\, e^{-j\omega k}e^{-j\omega m}$$

$$= \sum_{k=-\infty}^{\infty} h(k)e^{-j\omega k}\sum_{m=-\infty}^{\infty} x(m)\, e^{-j\omega m} = H(e^{j\omega})X(e^{j\omega})$$

设 $X(z)=\mathrm{ZT}[x(n)]$ 与 $H(z)=\mathrm{ZT}[h(n)]$ 的收敛域分别为 (R_{x-},R_{x+}) 与 (R_{h-},R_{h+})，则也类似地存在 $Y(z)=\mathrm{ZT}[y(n)]=X(z)H(z)$，新的收敛域是 $(\max(R_{x-},R_{h-}),\min(R_{x+},R_{h+}))$。因此对由式(14.7)描述的一般性 LTI 系统，存在

$$H(z) = \frac{Y(z)}{X(z)} = \frac{\sum_{m=0}^{M} b_m z^{-m}}{1-\sum_{m=1}^{N} a_m z^{-m}} \tag{14.16}$$

若用 ZT2[・]表示 2 维 ZT，对由式(14.9)描述的二维 LTI 系统类似地有

$$Y(z_1,z_2) = H(z_1,z_2)X(z_1,z_2) \tag{14.17}$$

其中，$Y(z_1,z_2)=\mathrm{ZT2}[y(n_1,n_2)]$，$X(z_1,z_2)=\mathrm{ZT2}[x(n_1,n_2)]$，$H(z_1,z_2)=\mathrm{ZT2}[h(n_1,n_2)]$。对由式(14.10)描述的二维系统，其系统函数为

$$H(z_1,z_2) = \frac{Y(z_1,z_2)}{X(z_1,z_2)} = \frac{\sum_{m=0}^{M}\sum_{n=0}^{N} b_{mn} z_1^{-m} z_2^{-n}}{1-\sum_{m=1}^{M}\sum_{n=1}^{N} a_{mn} z_1^{-m} z_2^{-n}} \tag{14.18}$$

14.1.3　时域离散平稳随机信号及其统计描述

平稳随机信号指统计特性不随时间变化但不能用明确数学关系描述的信号。这类信号不但常见，并且将随机信号近似看作平稳随机的，可以简化对一些信号的处理与分析。

严格定义时域离散平稳随机信号需要先描述它的概率分布函数。时域离散随机信号可以被表示为不同时间点上的随机变量序列 $x(n)$，$n=1,2,\cdots$。若 $x(n)$ 幅值连续，它的概率分布函数为

$$F_x(s,n) = \Pr(x(n)\leqslant s(n)) \tag{14.19}$$

其中，$s(n)$ 是 $x(n)$ 的样本值；Pr（・）表示概率。若 $s(n)$ 对不同 n 均相同，可以将上述分布函数简化表示为 $F_x(s,n)=\Pr(x(n)\leqslant s)$。在以上情况下均有

$$p_x(s,n) = \frac{\mathrm{d}F_x(s,n)}{\mathrm{d}s},\quad F_x(s,n) = \int_s p_x(s,n)\mathrm{d}s \tag{14.20}$$

前者是后者的概率密度。若 $x(n)$ 的可能取值是离散的 $a_1,a_2,\cdots$，则可用分布率

$$\Pr(x(n) = a_i),\quad i = 1,2,\cdots \tag{14.21}$$

描述 $x(n)$ 的随机特征。若同时观测 $x(n)$ 中的 N 个连续变量 $x^N=(x(n_1),\cdots,x(n_N))$，则 x^N 的分布函数为

$$F_{x^N}(s_1,n_1,\cdots,s_N,n_N) = \Pr(x(n_1)\leqslant s_1,\cdots,x(n_N)\leqslant s_N) \tag{14.22}$$

概率密度为

$$p_{x^N}(s_1,n_1,\cdots,s_N,n_N) = \frac{\partial^N F_{x^N}(s_1,n_1,\cdots,s_N,n_N)}{\partial s_1 \partial s_2 \cdots \partial s_N} \tag{14.23}$$

并且类似地有

$$\begin{aligned} &F_{x^N}(s_1,n_1,\cdots,s_N,n_N) \\ &=\int_{s_1}\int_{s_2}\cdots\int_{s_N} p_{x^N}(s_1,n_1,\cdots,s_N,n_N)\mathrm{d}s_1\mathrm{d}s_2\cdots\mathrm{d}s_N \end{aligned} \tag{14.24}$$

若 $x(n_1),\cdots,x(n_N)$的可能取值分别是离散的$(a_{11},a_{12},\cdots),\cdots,(a_{N1},a_{N2},\cdots)$，则可用分布律

$$\Pr(x(n_1)=a_{1i_1},\cdots,x(n_N)=a_{Ni_N}),\quad i_1,\cdots,i_N=1,2,\cdots,N \tag{14.25}$$

描述 x^N 的随机特征。

定义 14.1.7 对时域离散随机信号 $x(n)$，$n=1,2,\cdots$，设 $N\geqslant 1$ 为任意不大于变量个数的正整数，如果

$$\begin{aligned} &F_{x^N}(s_1,n_1+m,\cdots,s_N,n_N+m) \\ &=F_{x^N}(s_1,n_1,\cdots,s_N,n_N),\quad n_1,\cdots,n_N,m\in\mathbf{Z} \end{aligned} \tag{14.26}$$

则 $x(n)$是平稳随机的。

由于式(14.26)要求的条件在实际情况下很难满足，由定义 14.1.7 给出的平稳随机信号通常被称为严平稳的。许多随机信号不是严平稳的，而被称为宽平稳的，设 $E(\cdot)$表示数学期望，$x(n)$为宽平稳的时域离散随机信号，它的均值和方差不随时间改变，这可分别表示为

$$m_x = E(x(n)) = E(x(n+m)) \tag{14.27}$$

$$\begin{aligned} \sigma_x^2 &= \mathrm{Var}(x(n)) = E(|x(n)-m_x|^2) \\ &= E(|x(n+m)-m_x|^2) \end{aligned} \tag{14.28}$$

同时，$x(n)$的自相关函数、自协方差函数以及与同类信号 $y(n)$的互相关函数均是时间差 m 的函数，这可分别表示为

$$\phi_{xx}(n,n+m)=E(x^*(n)x(n+m))\triangleq\phi_{xx}(m) \tag{14.29}$$

$$\mathrm{cov}_{xx}(n,n+m)=E\,((x(n)-m_x)^*(x(n+m)-m_x))\triangleq\mathrm{cov}_{xx}(m) \tag{14.30}$$

$$\phi_{xy}(n,n+m)=E(x^*(n)y(n+m))\triangleq\phi_{xy}(m) \tag{14.31}$$

其中，上标“*”表示共轭；$\triangleq$表示“记为”。显然，将信号作为宽平稳的将有助于简化相关的处理和分析。

功率谱密度(亦简称功率谱)也是刻画宽平稳时域离散随机信号的重要数字特征，它与自相关函数分别从频域和时域反映了信号的二阶统计特性。对以上 $x(n)$，它的功率谱密度可以表示为

$$\Phi_{xx}(\mathrm{e}^{\mathrm{j}\omega}) = \sum_{m=-\infty}^{\infty}\phi_{xx}(m)\,\mathrm{e}^{-\mathrm{j}\omega m} \tag{14.32}$$

即 $x(n)$功率谱密度是其自相关函数的傅里叶变换，则也有

$$\phi_{xx}(m) = \frac{1}{2\pi}\int_{-\pi}^{\pi}\Phi_{xx}(\mathrm{e}^{\mathrm{j}\omega})\,\mathrm{e}^{-\mathrm{j}\omega m}\,\mathrm{d}\omega \tag{14.33}$$

LTI 系统常用于处理平稳时域离散随机信号，关于它对相关统计特性的影响有以下的定理。

定理 14.1.3　设 LTI 系统的单位冲激响应为 $h(n)$，频率响应为 $H(\mathrm{e}^{\mathrm{j}\omega})$，输入 $x(n)$是宽平稳时域离散随机信号，$\phi_{xx}(m)$与 $\Phi_{xx}(\mathrm{e}^{\mathrm{j}\omega})$分别是它的自相关函数和功率谱密度，若

(1) $\phi_{xx}(m)$与 $h(n)$绝对可和；

(2) $\sum_{k=-\infty}^{\infty}\sum_{n=-\infty}^{\infty}h^*(k)h(n)\phi_{xx}(n-k)<\infty$；

则有：

(1) 输出 $y(n)=h(n)*x(n)$也是宽平稳的；

(2) $y(n)$的均值和相关函数分别为

$$m_y = m_x\sum_{n=-\infty}^{\infty}h(n) \tag{14.34}$$

$$\phi_{yy}(n,n+m) = \sum_{k=-\infty}^{\infty}h(k)\sum_{r=-\infty}^{\infty}h(r)\phi_{yy}(m+k-r) \triangleq \phi_{xx}(m) \tag{14.35}$$

(3) $y(n)$的功率谱密度为

$$\Phi_{yy}(\mathrm{e}^{\mathrm{j}\omega}) = |H(\mathrm{e}^{\mathrm{j}\omega})|^2\Phi_{xx}(\mathrm{e}^{\mathrm{j}\omega}) \tag{14.36}$$

例 14.1.3　理想白噪声在各个频率上的能量相等，因此是一类具有常数功率谱密度

$$\Phi_{xx}(\mathrm{e}^{\mathrm{j}\omega}) = \sigma_x^2 \qquad \text{对全部 } \omega$$

的信号，其中 σ_x 为常数。若定义白噪声 $x(n)$的功率为 $E(x^2(n))$，则

$$E(x^2(n)) = \phi_{xx}(0) = \frac{1}{2\pi}\int_{-\pi}^{\pi}\Phi_{xx}(\mathrm{e}^{\mathrm{j}\omega})\mathrm{d}\omega = \frac{1}{2\pi}\int_{-\pi}^{\pi}\sigma_x^2\mathrm{d}\omega = \sigma_x^2$$

因此白噪声为常数功率。若使白噪声通过一个未知的系统，根据式(14.36)有

$$\Phi_{yy}(\mathrm{e}^{\mathrm{j}\omega}) = |H(\mathrm{e}^{\mathrm{j}\omega})|^2\sigma_x^2$$

其中，$\Phi_{yy}(\mathrm{e}^{\mathrm{j}\omega})$是输出的功率谱密度；$H(\mathrm{e}^{\mathrm{j}\omega})$是未知的频率响应。由于 $\Phi_{yy}(\mathrm{e}^{\mathrm{j}\omega})$可通过输出计算，因此在白噪声输入下$|H(\mathrm{e}^{\mathrm{j}\omega})|^2$ 容易被估计，这是辨识系统的常用方法之一。

14.1.4　信号质量评价

由于不可避免地引入了噪声，信息隐藏会影响载体信号的感知质量。为衡量变化的程度，当前普遍采用主观评价和客观评价两类方法，前者主要依靠人的感知系统(视觉和听觉)进行评价，后者需要对信号进行计算。以下以尺寸为 $M\times N$ 的 2 维信号为例给出几个常用评价标准的计算方法及其相互关系：

(1) 均方差(MSE)

$$\mathrm{MSE} = \frac{1}{MN}\sum_{m=1}^{M}\sum_{n=1}^{N}(I(m,n)-I'(m,n))^2 \tag{14.37}$$

其中，$I(m,n)$为原始信号；$I'(m,n)$是引入噪声后的信号。

(2) 信噪比(SNR)

$$\mathrm{SNR} = 10\lg\frac{\sum_{m=1}^{M}\sum_{n=1}^{N}I(m,n)^2}{\sum_{m=1}^{M}\sum_{n=1}^{N}(I(m,n)-I'(m,n))^2}$$

$$=10\lg \frac{\sum_{m=1}^{M}\sum_{n=1}^{N} I(m,n)^2}{MN \cdot \text{MSE}} \quad (\text{dB}) \tag{14.38}$$

(3) 峰值信噪比(PSNR)

$$\text{PSNR} = 10\lg \frac{MN \cdot P^2}{\sum_{m=1}^{M}\sum_{n=1}^{N}(I(m,n)-I'(m,n))^2}$$

$$=10\lg \frac{P^2}{\text{MSE}} \quad (\text{dB}) \tag{14.39}$$

其中 P 是信号的峰值,即最大可取的值。

14.2 信号变换

以上介绍的傅里叶变换和 Z 变换常用于信号分析,而本章介绍的变换常用于对实际信号的处理或编码,它们的变换系数也形成了信息隐藏最基本的嵌入域。

14.2.1 离散傅里叶变换

定义 14.2.1 设 $x(n)$为长度为 M 的有限长时域离散信号序列,则它的 N 点离散傅里叶变换(DFT)为

$$X(k) = \text{DFT}[x(n)] = \sum_{n=0}^{N-1} x(n)\mathrm{e}^{-\mathrm{j}\frac{2\pi}{N}kn}, \quad 0 \leqslant k \leqslant N-1 \tag{14.40}$$

其中,$N \geqslant M$;k 为整数。

令

$$x(n) = \text{IDFT}[X(k)] = \frac{1}{N}\sum_{k=0}^{N-1} X(k)\,\mathrm{e}^{\mathrm{j}\frac{2\pi}{N}kn}, \quad 0 \leqslant n \leqslant N-1 \tag{14.41}$$

可验证式(14.41)是以上情况下的逆 DFT(IDFT)。

DFT 是典型的正交变换,若将式(14.40)与式(14.41)分别写为 $\boldsymbol{X}=\boldsymbol{F}\boldsymbol{X}$ 与 $\boldsymbol{X}=\boldsymbol{F}^{-1}\boldsymbol{X}$ 的矩阵形式,则变换矩阵 $\boldsymbol{F}$ 满足 $\boldsymbol{F}^{\mathrm{T}}=\boldsymbol{F}^{-1}$,其中 $\boldsymbol{F}^{\mathrm{T}}$ 是 $\boldsymbol{F}$ 的转置。

根据式(14.40)和式(14.13),可知 DFT 与 Z 变换的关系为

$$X(k) = X(z)\mid_{\mathrm{e}^{\mathrm{j}\frac{2\pi}{N}kn}} \tag{14.42}$$

DFT 主要还包括以下的性质。

性质 14.2.1(线性性质) 若 $x_1(n)$与 $x_2(n)$的长度分别为 N_1 与 N_2,取 $N=\max(N_1,N_2)$,则

$$\text{DFT}[ax_1(n)+bx_2(n)] = a\text{DFT}[x_1(n)]+b\text{DFT}[x_2(n)] \tag{14.43}$$

其中 a 与 b 为常数。

性质 14.2.2(周期性) $X(k)=\text{DFT}(x(n))$与 $x(n)=\text{IDFT}(X(k))$均以 N 为周期,即对任意整数 m,在式(14.40)中有 $X(k)=X(k+mN)$,在式(14.41)中有 $x(n)=x(n+mN)$,它在整个时间轴上相当于对原输入信号进行了周期为 N 的延

拓，记为 $\tilde{x}(n)=x\,(n)_N$，原输入的长度为 N 的信号可以表示为 $x(n)=\tilde{x}(n)R_N(n)$，其中，$R_N(n)$ 为长度为 N 的矩形信号。

性质 14.2.3（时域循环移位定理）　若 $y(n)=\tilde{x}(n+m)R_N(n)$，则

$$Y(k) = \mathrm{DFT}[y(n)] = \mathrm{e}^{-km\frac{2\pi}{N}\mathrm{j}}X(k),\quad 0 \leqslant k \leqslant N-1 \tag{14.44}$$

性质 14.2.4（频域循环移位定理）　若 $Y(k)=\tilde{X}(k+m)R_N(k)$，则

$$y(n) = \mathrm{IDFT}\ [Y(k)] = \mathrm{e}^{nm\frac{2\pi}{N}\mathrm{j}}x(n),\quad 0 \leqslant k \leqslant N-1 \tag{14.45}$$

其中 $\tilde{X}(k)$ 是 $X(k)$ 以 N 为周期的延拓。

性质 14.2.5（循环卷积定理）　若 $X(k)=X_1(k)X_2(k)$，$X_1(k)=\mathrm{DFT}[x_1(n)]$，$X_2(k)=\mathrm{DFT}[x_2(n)]$，$\tilde{x}_1(n)=x_1\,(n)_N$，$\tilde{x}_2(n)=x_2\,(n)_N$，则

$$\begin{aligned} x(n) = \mathrm{IDFT}[X(k)] &= \sum_{m=0}^{N} x_1(m)\tilde{x}_2(n-m)R_N(n) \\ &= \sum_{m=0}^{N} x_2(m)\tilde{x}_1(n-m)R_N(n) \triangleq x_1(n)\otimes x_2(n) \end{aligned} \tag{14.46}$$

性质 14.2.6（能量守恒）　若 $X(k)=\mathrm{DFT}(x(n))$，则

$$\sum_{n=0}^{N-1} |\,x(n)\,|^2 = \frac{1}{N}\sum_{k=0}^{N-1} |\,X(k)\,|^2 \tag{14.47}$$

性质 14.2.7（共轭对称性）　若 $X(k)=\mathrm{DFT}(x(n))$，则 $X(k)=X^*(N-k)$。

例 14.2.1　以上 DFT 的很多性质都与它隐含的周期性相关，这一性质可以用一个周期信号的离散傅里叶级数(DFS)表示来阐明。若式(14.40)中的 $x(n)$ 为周期信号，则该式与式(14.41)分别被称为 DFS 的分析式和综合式。取周期信号

$$\tilde{x}(n) = \sum_{r=-\infty}^{\infty}\delta(n-rN) = \begin{cases} 1 & n = rN \\ 0 & n \neq rN \end{cases}$$

其中 r 为任意整数。取 $\tilde{x}(n)$ 的一个周期进行 DFS 分析：

$$\tilde{X}(k) = \sum_{n=0}^{N-1}\delta(n)\,\mathrm{e}^{-\mathrm{j}\frac{2\pi}{N}kn} = \mathrm{DFT}\left[\tilde{x}(n)\,|_{0\leqslant n\leqslant N-1}\right] = \mathrm{e}^0 = 1$$

基于上述变换系数，将 $\tilde{x}(n)$ 表示为

$$\tilde{x}(n) = \sum_{r=-\infty}^{\infty}\delta(n-rN) = \frac{1}{N}\sum_{r=0}^{N-1}\mathrm{e}^{\mathrm{j}\frac{2\pi}{N}kn} = \mathrm{IDFT}[\tilde{X}(k)]$$

因此，对 $\tilde{x}(n)$ 中一个周期进行 DFT 和 IDFT，最后得到的 IDFT 输出正是原来的周期函数。

信息隐藏算法可以将信息嵌入 DFT 系数的幅度和相位中，但根据性质 14.2.6，这种嵌入不能改变系数的共轭对称性，否则 IDFT 不能保证将复数输入映射到实数域。

设 $x(n_1,n_2)$ 为包含 $N_1\times N_2$ 个样点的二维信号，则二维 DFT 和 IDFT 可以分别表示为

$$\begin{aligned} X(k_1,k_2) &= \mathrm{DFT2}[x(n_1,n_2)] \\ &= \sum_{n_1=0}^{N_1-1}\sum_{n_2=0}^{N_2-1} x(n_1,n_2)\,\mathrm{e}^{-\mathrm{j}\frac{2\pi}{N_1}k_1n_1}\,\mathrm{e}^{-\mathrm{j}\frac{2\pi}{N_2}k_2n_2} \end{aligned} \tag{14.48}$$

$$x(n_1,n_2) = \mathrm{IDFT2}[X(k_1,k_2)] = \frac{1}{N_1 N_2}\sum_{k_1=0}^{N_1-1}\sum_{k_2=0}^{N_2-1} X(k_1,k_2)\mathrm{e}^{\mathrm{j}\frac{2\pi}{N}k_1 n_1}\mathrm{e}^{\mathrm{j}\frac{2\pi}{N}k_2 n_2} \tag{14.49}$$

其中 k_1 与 k_2 为整数，并且 $0 \leqslant k_1 \leqslant N_1-1$，$0 \leqslant k_2 \leqslant N_2-1$。可以看出，二维 DFT 和 IDFT 的变换核都是可分离的，式 (14.48)与式(14.49)可以分别写为

$$\boldsymbol{X}_{N_1\times N_2} = \boldsymbol{F}_{N_1\times N_1}\boldsymbol{x}_{N_1\times N_2}\boldsymbol{F}^{\mathrm{T}}_{N_2\times N_2} \tag{14.50}$$

$$\boldsymbol{x}_{N_1\times N_2} = \boldsymbol{F}^{\mathrm{T}}_{N_1\times N_1}\boldsymbol{X}_{N_1\times N_2}\boldsymbol{F}^{\mathrm{T}}_{N_2\times N_2} \tag{14.51}$$

14.2.2 离散余弦变换

与以上 DFT 不同，离散余弦变换(DCT)及其逆变换(IDCT)均是实数域之间的映射。

定义 14.2.2 设 $x(n)$是长度为 N 的有限长时域离散信号序列，则它的 DCT 为

$$X(k) = \mathrm{DCT}[x(n)] = \sqrt{\frac{2}{N}}\sum_{n=0}^{N-1} c(k)x(n)\cos\frac{2n+1}{2N}\pi k \tag{14.52}$$

其中 $c(0)=1/\sqrt{2}$，当 $k=1,2,\cdots,N-1$ 有 $c(k)=1$。

式(14.52)右侧中的每个求和项可以写成不同切比雪夫多项式与相应 $x(n)$的乘积，则根据切比雪夫多项式的正交性，可验证 $x(n)$的 IDCT 为

$$x(n) = \mathrm{IDCT}[X(k)] = \sqrt{\frac{2}{N}}\sum_{k=0}^{N-1} c(k)X(k)\cos\frac{2n+1}{2N}\pi k \tag{14.53}$$

DCT 和 DFT 都是正交变换，均常用于对信号进行频域分析。以下定理反映了它们内在的联系。

定理 14.2.1 若将 $x(n)$延拓为

$$x_e(n) = \begin{cases} x(n) & n = 0,1,\cdots,N-1 \\ 0 & n = N,N+1,\cdots,2N-1 \end{cases} \tag{14.54}$$

则对 $x(n)$的 N 点 DCT 可以表示为

$$X(k) = \mathrm{DCT}[x(n)] = \sqrt{\frac{2}{N}}\mathrm{Re}\{\mathrm{e}^{-\mathrm{j}\frac{k\pi}{2N}}\mathrm{DFT}[x_e(n)]\},\quad k = 0,1,\cdots,N-1 \tag{14.55}$$

其中，$\mathrm{DFT}[x_e(n)]$为对 $x_e(n)$的 $2N$ 点 DFT，$\mathrm{Re}\{\cdot\}$表示取实部。

证明： 根据式(14.52)和式(14.54)有

$$\begin{aligned} X_e(k) &= \mathrm{DCT}[x_e(n)] = \sqrt{\frac{2}{N}}\sum_{n=0}^{2N-1} c(k)x_e(n)\mathrm{Re}\{\mathrm{e}^{-\mathrm{j}\frac{2n+1}{2N}\pi k}\} \\ &= \sqrt{\frac{2}{N}}\mathrm{Re}\left\{\sum_{n=0}^{2N-1} c(k)x_e(n)\mathrm{e}^{-\mathrm{j}\frac{2n+1}{2N}\pi k}\right\} \\ &= \sqrt{\frac{2}{N}}\mathrm{Re}\left\{\mathrm{e}^{-\mathrm{j}\frac{\pi k}{2N}}c(k)\sum_{n=0}^{2N-1} x_e(n)\mathrm{e}^{-\mathrm{j}\frac{2n\pi k}{2N}}\right\} \\ &= \sqrt{\frac{2}{N}}\mathrm{Re}\{\mathrm{e}^{-\mathrm{j}\frac{\pi k}{2N}}c(k)\mathrm{DFT}[x_e(n)]\} \end{aligned} \tag{14.56}$$

当 $k=0,1,\cdots,N-1$，显然有 $\mathrm{DCT}[x_e(n)]=\mathrm{DCT}[x(n)]$，因此定理得证。

若将 $X(k)$ 延拓为

$$X_e(k)=\begin{cases}X(k) & k=0,1,\cdots,N-1\\0 & k=N,N+1,\cdots,2N-1\end{cases}\tag{14.57}$$

还可以类似地证明

$$\begin{aligned}x(n)&=\mathrm{IDCT}[X(k)]=\mathrm{IDCT}[X_e(k)]\\&=\sqrt{\frac{2}{N}}\sum_{k=0}^{N}c(k)X_e(k)\mathrm{Re}\left\{\mathrm{e}^{-\mathrm{j}\frac{2n+1}{2N}\pi k}\right\}\\&=\sqrt{\frac{2}{N}}\mathrm{Re}\left\{c(k)\sum_{k=0}^{N}(X_e(k)\mathrm{e}^{-\mathrm{j}\frac{\pi k}{2N}})\ \mathrm{e}^{-\mathrm{j}\frac{2n\pi k}{2N}}\right\}\\&=\sqrt{\frac{2}{N}}\mathrm{Re}\left\{c(k)\mathrm{DFT}[X_e(k)\mathrm{e}^{-\mathrm{j}\frac{\pi k}{2N}}]\right\}\end{aligned}\tag{14.58}$$

DCT 的以上性质不但揭示了它与 DFT 的内在联系，还说明计算 DCT 可以通过计算 DFT 得到。由于 DFT 存在快速算法 FFT(Fast Fourier Transform)，这使计算 DCT 也存在相应的快速算法，促进了它的应用。

设 $x(n_1,n_2)$ 为包含 $N_1\times N_2$ 个样点的 2 维信号，2 维 DCT 和 IDCT 可以分别表示为

$$\begin{aligned}X(k_1,k_2)&=\mathrm{DCT2}[x(n_1,n_2)]\\&=\frac{2}{\sqrt{N_1N_2}}\sum_{n_1=0}^{N_1-1}\sum_{n_2=0}^{N_2-1}c_1(k_1)c_2(k_2)x(n_1,n_2)\\&\quad\cos\left(\frac{2n_1+1}{2N_1}k_1\pi\right)\cos\left(\frac{2n_2+1}{2N_2}k_2\pi\right)\end{aligned}\tag{14.59}$$

$$\begin{aligned}x(n_1,n_2)&=\mathrm{IDCT2}[X(k_1,k_2)]\\&=\frac{2}{\sqrt{N_1N_2}}\sum_{k_1=0}^{N_1-1}\sum_{k_2=0}^{N_2-1}c_1(k_1)c_2(k_2)X(k_1,k_2)\\&\quad\cos\left(\frac{2n_1+1}{2N_1}k_1\pi\right)\cos\left(\frac{2n_2+1}{2N_2}k_2\pi\right)\end{aligned}\tag{14.60}$$

其中，$c_1(0)=c_2(0)=1/\sqrt{2}$，当 k_1 或 $k_2=1,2,\cdots,N-1$，有 $c_1(k)=c_2(k)=1$。另外，一维和二维DCT 也存在与 DFT 类似的矩阵表示。

14.4.1 小节将给出 2 维图像 DCT 变换的实例。

14.2.3　离散时间小波多分辨率分解

与前述的信号变换不同，离散小波变换[5]同时反映了信号的时域和频域特性，因此其输出被称为时频域。

定义 14.2.3　设 $x(t)$ 为连续时间信号，小波函数 $\psi_{j,k}(t)$ 定义在有限区域内，其中 j 与 k 是两个整数参数，它们可将 $x(t)$ 展开为

$$x(t)=\sum_{j}\sum_{k}a_j(k)\psi_{j,k}(t),\quad j,k\in Z\tag{14.61}$$

则计算 $a_j(k)=\langle x(t),\psi_{j,k}(t)\rangle$ 就是对 $x(t)$ 进行离散小波变换(DWT)，其中

$<\cdot,\cdot>$表示计算内积；若 t 为离散的，则变换为离散时间小波变换(DTWT)。

值得注意的是，DWT 仅仅参数是离散的。常用的小波函数有以下形式，即

$$\psi_{j,k}(t) = 2^{j/2}\psi(2^j t - k) \tag{14.62}$$

具有以上形式的小波函数常被称为二进小波，通过参数的变化，它们构成了小波变换的基函数集合，其中 $\psi(t)$ 被称为母小波函数。

DTWT 一般不直接使用，而是在离散时间小波多分辨率分解(WMRA)中发挥作用。后者是最常用的时频处理和分析方法之一，它的输出不但包含小波系数，也包括所谓的尺度系数。以下将逐步介绍这些概念及其关系，为方便论述，这里先基于 DWT 和函数空间分解的概念引出 WMRA，再由后者给出 DTWT 通常的使用和计算方法。设 $L^2(\mathbf{R})$代表 2 次可积函数空间，DWT 可以将任意 $x(t)\in L^2(\mathbf{R})$分解为由加号连接的两部分，即

$$x(t) = \sum_{j<J}\sum_k a_j(k)\psi_{j,k}(t) + \sum_{j\geqslant J}\sum_k a_j(k)\psi_{j,k}(t) \tag{14.63}$$

由于前一部分的小波频率较低，因此被称为概貌信息，这样，第 2 部分相应地被称为细节信息。由于小波函数 $\psi_{j,k}(t)$待定，不妨取 $J=0$，而由于它们是不同的基函数，因此存在

$$V_0 = \overline{\mathop{\mathrm{Span}}_{j<0,k}\{\psi_{j,k}(t)\}},\quad W_j = \overline{\mathop{\mathrm{Span}}_{k}\{\psi_{j,k}(t)\}}\ \text{且}\ j \geqslant 0 \tag{14.64}$$

其中，V_0 为全部 $\psi_{j<0,k}(t)$张成的空间；W_j 为 $\psi_{j,k}(t)$张成的空间，其中 $j\geqslant 0$，也即这些空间中任意函数可按相应参数下的小波基函数展开；由于 $x(t)\in L^2(\mathbf{R})$，式(14.63)蕴含着

$$V_0\oplus W_0\oplus W_1\oplus\cdots = L^2(\mathbf{R}) \tag{14.65}$$

$$V_j\oplus W_j = V_{j+1}\quad j\geqslant 0 \tag{14.66}$$

$$\cdots\oplus W_{-2}\oplus W_{-1}\oplus W_0\oplus W_1\oplus W_2\oplus\cdots = L^2(\mathbf{R}),\quad V_{-\infty}=\varnothing \tag{14.67}$$

$$\cdots\subset V_{-2}\subset V_{-1}\subset V_0\subset V_1\subset V_2\subset\cdots\subset L^2(\mathbf{R}) \tag{14.68}$$

其中，$\oplus$为函数空间的直和；$\varnothing$表示空集。另取

$$V_0 = \overline{\mathop{\mathrm{Span}}_{j<J,k}\{\psi_{j,k}(t)\}} = \overline{\mathop{\mathrm{Span}}_{k}\{\varphi_{0,k}(t)\}} \tag{14.69}$$

其中，$\varphi_{0,k}(t)$被称为在尺度为零并且移位为 k 下的尺度函数(以下将给出其存在条件)，使它们仅通过移位即可张成 V_0，则 $x(t)$可以展开为

$$x(t) = \sum_k c_0(k)\varphi_{0,k}(t) + \sum_{j\geqslant 0}\sum_k d_j(k)\psi_{j,k}(t),\quad j,k\in\mathbf{Z} \tag{14.70}$$

其中，$c_0(k)$为尺度 0 下的尺度系数；$d_j(k)$为尺度 j 下的小波系数。不同尺度下的尺度系数和小波系数构成了 WMRA 的输出。以上分析说明，WMRA 的输出包含了 DWT 在部分尺度下的输出。

构造 WMRA 需要确定尺度函数，而尺度函数也能确定小波函数。设待定的尺度函数也有

$$\varphi_{j,k}(t) = 2^{j/2}\varphi(2^j t - k) \tag{14.71}$$

的形式，由于 $V_0\subset V_1$，V_0 中的函数 $\varphi_{0,0}(t)=\varphi(t)$与 $\psi_{0,0}(t)=\psi(t)$均分别可以由 V_1 中的基函数 $\varphi_{1,k}(t)=\sqrt{2}\,\varphi(2t-k)$生成，则存在以下关系：

$$\varphi(t) = \sum_n h_0(n)\sqrt{2}\,\varphi(2t-n) \tag{14.72}$$

$$\psi(t)=\sum_n h_1(n)\sqrt{2}\varphi(2t-n) \tag{14.73}$$

以及以下定理。

定理 14.2.2　尺度系数 $c_j(k)$满足

$$c_j(k)=\sum_m h_0(m-2k)c_{j+1}(m) \tag{14.74}$$

证明：由式(14.71)和式(14.72)得到

$$\begin{aligned}\varphi(2^jt-k)&=\sum_n h_0(n)\sqrt{2}\varphi(2(2^jt-k)-n)\\&=\sum_n h_0(n)\sqrt{2}\varphi(2(2^jt-k)-n)\end{aligned}$$

令 $m=2k+n$,上式变为

$$\varphi(2^jt-k)=\sum_m h_0(m-2k)\sqrt{2}\varphi(2^{j+1}t-m) \tag{14.75}$$

由于

$$V_j=\mathop{\mathrm{Span}}_k\{2^{j/2}\varphi(2^jt-k)\}$$

若 $f(t)\in V_{j+1}$,则可以在其中展开为

$$f(t)=\sum_k c_{j+1}(k)2^{(j+1)/2}\varphi(2^{j+1}t-k)$$

由于 $V_{j+1}=V_j+W_j$,$f(t)$也可展开为

$$f(t)=\sum_k c_j(k)2^{j/2}\varphi(2^jt-k)+\sum_k d_j(k)2^{j/2}\psi(2^jt-k)$$

由于

$$c_j(k)=\langle f(t),\varphi_{j,k}(t)\rangle=\int f(t)2^{j/2}\varphi(2^jt-k)\mathrm{d}t$$

根据式(14.75)有

$$c_j(k)=\sum_m h_0(m-2k)\int f(t)2^{(j+1)/2}\varphi(2^{j+1}t-k)\mathrm{d}t=\sum_m h_0(m-2k)c_{j+1}(m)$$

类似地可以证明以下定理。

定理 14.2.3　小波系数 $d_j(k)$满足

$$d_j(k)=\sum_m h_1(m-2k)c_{j+1}(m) \tag{14.76}$$

定理 14.2.4　尺度系数 $c_j(k)$与小波系数 $d_j(k)$满足

$$c_{j+1}(k)=\sum_m c_j(m)h_0(k-2m)+\sum_m d_j(m)h_1(k-2m) \tag{14.77}$$

定理 14.2.5　若 V_0 与 W_0 互为正交补空间,即

$$\int\varphi(t-n)\psi(t-m)\mathrm{d}t=0 \tag{14.78}$$

则

$$h_1(n)=\pm(-1)^n h_0(N-n) \tag{14.79}$$

其中 N 为任意选择的奇数,并且

$$\sum_n h_0(n)h_1(n-2k)=0 \tag{14.80}$$

根据定理14.2.2和定理14.2.3，可以仅构造两个系数分别是$h_0(-n)$与$h_1(-n)$的滤波器（分别称为尺度滤波器和小波滤波器，或称为小波滤波器组）实现离散时间WMRA。方法是，令输入数据$c_j(n)=x(n)$，随后逐级（stage）分解，这也是DTWT最常用的计算和使用方法。图14.1给出了一个两级双通道（band）WMRA的计算流程，其中也附带给出了函数空间的变化，↓2表示下采样（或称二抽取），即每隔一个样点取样一次，而舍弃跳过的样点。

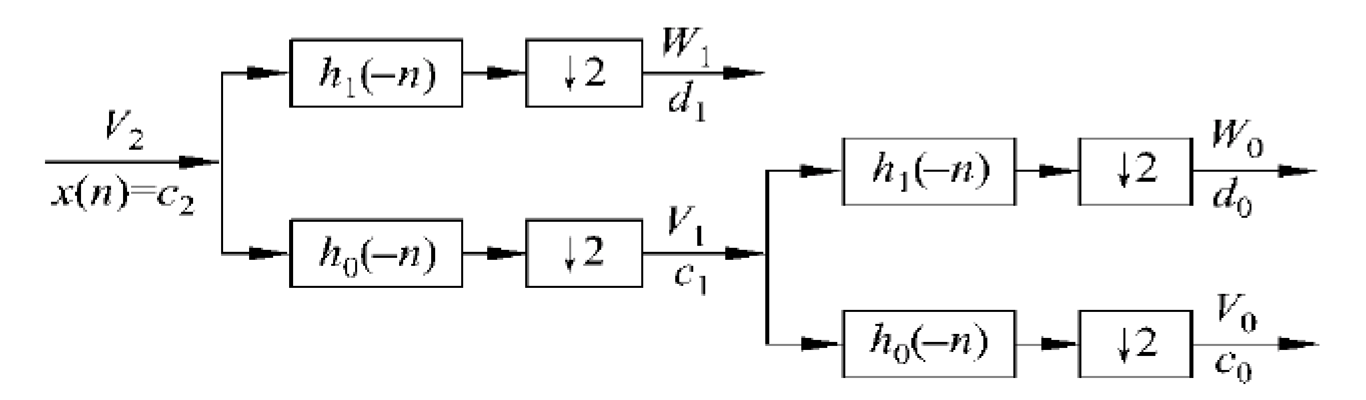


图14.1　两级双通道WMRA的计算流程

根据定理14.2.4，可以仅仅构造两个系数分别是$h_0(n)$与$h_1(n)$的滤波器实现离散时间WMRA系数的综合。方法是，逐级输入数据$c_j(n)$与$d_j(n)$并计算$c_{j+1}(n)$即可，这也是逆DTWT最常用的计算和使用方法。图14.2给出了一个两级双通道WMRA系数的综合计算流程，其中也附带给出了函数空间的变化，↑2表示上采样（或称二插值），即每隔一个样点插入一个样点。

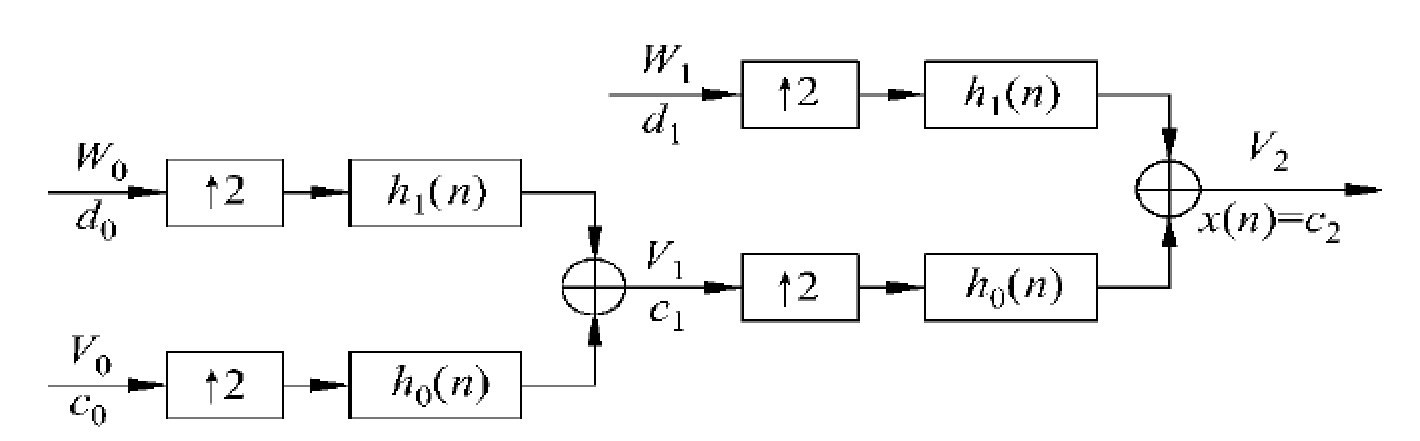


图14.2　两级双通道WMRA综合的计算流程

以上分析结果进一步简化了WMRA系统的构造，它将问题仅归结为获得系数$h_0(n)$。研究表明，$h_0(n)$需要满足一些必要的性质，因此，可以通过构造或求解满足这些性质的$h_0(n)$得到WMRA（见表14.1）。$h_0(n)$基本的性质包括：

性质14.2.8（规整性）　$h_0(n)$应满足$\sum_n h_0(n)=1$。

性质14.2.9（双正交性）　$h_0(n)$应满足$\sum_n h_0(n)h_0(n-2k)=0, k\neq 0$。

性质14.2.10（低通性）　$h_0(n)$应满足$\sum_n (-1)^n h_0(n)=0$。

表14.1　部分Daubechies小波滤波器组对应的系数$h_0(n)$

长度	$h_0(0)$	$h_0(1)$	$h_0(2)$	$h_0(3)$	$h_0(4)$	$h_0(5)$	$h_0(6)$	$h_0(7)$	$h_0(8)$	$h_0(9)$
4	0.4830	0.8365	0.2241	−0.1294	/	/	/	/	/	/
6	0.3327	0.8069	0.4599	−0.1350	−0.0854	0.0352	/	/	/	/
8	0.2304	0.7148	0.6309	−0.0280	−0.1870	0.0308	0.0329	−0.0106	/	/
10	0.1601	0.6038	0.7243	0.1384	−0.2423	−0.0322	0.0776	−0.0062	−0.0126	−0.0033

在二维信号情况下，可以类似地将函数空间 V_{j+1} 划分为 $V_{j+1}=V_j\oplus W_j^1\oplus W_j^2\oplus W_j^3$，其中，$V_j$、$W_j^1$、$W_j^2$ 与 W_j^3 是相互正交的子空间，则

$$L^2(\mathbf{R}^2)=\bigoplus_{j\in\mathbf{Z}}(W_j^1\oplus W_j^2\oplus W_j^3),\quad V_{-\infty}=\varnothing \tag{14.81}$$

存在 2 维变量可分离的尺度函数

$$\varphi_{j,m,n}(x,y)=\varphi_{j,m}(x)\varphi_{j,n}(y) \tag{14.82}$$

它是 V_j 的正交基函数，$j,m,n\in\mathbf{Z}$，也存在变量可分离的小波函数

$$\psi_{j,m,n}^1(x,y)=\varphi_{j,m}(x)\psi_{j,n}(y) \tag{14.83}$$

$$\psi_{j,m,n}^2(x,y)=\psi_{j,m}(x)\varphi_{j,n}(y) \tag{14.84}$$

$$\psi_{j,m,n}^3(x,y)=\psi_{j,m}(x)\psi_{j,n}(y) \tag{14.85}$$

分别是 W_j^1、W_j^2 与 W_j^3 的正交基函数。则 $f(x,y)$ 可展开为

$$\begin{aligned}f(x,y)=&\sum_{k,l}c_0(k,l)\varphi_{0,k}(x)\varphi_{0,l}(y)+\sum_{j\geqslant0}\sum_{k,l}d_j^1(k,l)\varphi_{j,k}(x)\psi_{j,l}(y)\\&+\sum_{j\geqslant0}\sum_{k,l}d_j^2(k,l)\psi_{j,k}(x)\varphi_{j,l}(y)\\&+\sum_{j\geqslant0}\sum_{k,l}d_j^3(k,l)\psi_{j,k}(x)\psi_{j,l}(y)\end{aligned} \tag{14.86}$$

其中 $j,k,l\in\mathbf{Z}$。通常称 $c_j(k,l)$ 组成第 j 级 LL 子带，$d_j^1(k,l)$ 组成第 j 级 LH 子带，$d_j^2(k,l)$ 组成第 j 级 HL 子带，$d_j^3(k,l)$ 组成第 j 级 HH 子带。对二维 WMRA 相应地有

$$c_j(k,l)=\sum_{m,n}h_0(m-2k)h_0(n-2l)c_{j+1}(m,n) \tag{14.87}$$

$$d_j^1(k,l)=\sum_{m,n}h_0(m-2k)h_1(n-2l)c_{j+1}(m,n) \tag{14.88}$$

$$d_j^2(k,l)=\sum_{m,n}h_1(m-2k)h_0(n-2l)c_{j+1}(m,n) \tag{14.89}$$

$$d_j^3(k,l)=\sum_{m,n}h_1(m-2k)h_1(n-2l)c_{j+1}(m,n) \tag{14.90}$$

对以上二维 WMRA 的综合是

$$\begin{aligned}c_{j+1}(k,l)=&\sum_{m,n}c_j(m,n)h_0(k-2m)h_0(l-2n)\\&+\sum_{m,n}d_j^1(m,n)h_0(k-2m)h_1(l-2n)\\&+\sum_{m,n}d_j^2(m,n)h_1(k-2m)h_0(l-2n)\\&+\sum_{m,n}d_j^3(m,n)h_1(k-2m)h_1(l-2n)\end{aligned} \tag{14.91}$$

14.4.2 小节将给出二维图像小波信号变换的实例。

14.3 信号调制、嵌入与提取

信息隐藏一般通过编码、加密与(或)调制获得待隐藏信号，再通过替换、叠加、调制载体信号或其统计量等方法嵌入待隐藏信号。本节将介绍主要的调制、嵌入和提

取方法。

14.3.1 实值伪随机信号的产生

当信息隐藏算法不直接嵌入待隐藏信息编码的二进制串时，往往需要调制这个二进制串，调制所用的调制信号是实值伪随机信号。当一段实值伪随机信号代表一个特定的含义或实体，它也可以直接被嵌入。为使嵌入信号隐蔽，算法需要以上伪随机信号满足一些随机性质，主要是要求它们在一定范围内均匀分布或者正态分布。因此本小节以下将介绍产生(0,1)间均匀分布实值伪随机信号与获得正态分布伪随机信号的数学方法。需指出，由于计算误差等原因，完全均匀分布或者正态分布的信号是不能得到的，因此一般称通过计算设备获得的这类信号为伪随机信号。

获得(0,1)间均匀分布的实值伪随机信号具有极重要的理论和应用意义，根据这类信号序列可以产生具有任意概率分布的其他信号序列。以下定理通过指明任意分布和均匀分布随机变量之间的关系说明了这一点。

定理 14.3.1 设 ξ 是(0,1)间均匀分布的随机变量，记为 $\xi\sim\mathbf{U}(0,1)$，$F(x)$是任意分布函数，则

$$\eta = F^{-1}(\xi) \tag{14.92}$$

是服从 $F(x)$分布的随机变量，即 $F_\eta(x)\triangleq\Pr(\eta\leqslant x)=F(x)$。

证明： 由于 ξ 是(0,1)间均匀分布的随机变量，有

$$F_\xi(x) = \begin{cases} 0 & x\leqslant 0 \\ x & 0<x<1 \\ 1 & x\geqslant 1 \end{cases}$$

任何分布函数 $F(x)$均是单值、递增的非负函数，满足 $0\leqslant F(x)\leqslant 1$，因此$(\eta,\xi)$可视为 $F(x)$曲线上的一点。当取 $x=x_1$，有 $F_\eta(x_1)\triangleq\Pr(\eta\leqslant x_1)$，$\eta\leqslant x_1$ 等价于 $\xi\leqslant F(x_1)$，因此

$$F_\eta(x_1) = \Pr(\eta\leqslant x_1) = \Pr(\xi\leqslant F(x_1)) = F_\xi(F(x_1))$$

由于 $\xi\sim\mathbf{U}(0,1)$，当 $F(x_1)\in(0,1)$，有 $F_\xi(F(x_1))=F(x_1)$，则 $F_\eta(x_1)=F(x_1)$；当 $F(x_1)=0$，$F_\eta(x_1)=\Pr(\xi\leqslant 0)=0$，当 $F(x_1)=1$，$F_\eta(x_1)=\Pr(\xi\leqslant 1)=1$。因此 $F_\eta(x_1)=F(x_1)$在分布函数 $F(x_1)$与 $F_\eta(x_1)$的全部范围内得到满足，由于 x_1 是任意选定的，因此 $F_\eta(x)=F(x)$也成立。

有多种方法可以产生(0,1)间均匀分布的实值伪随机信号，它们包括线性同余法、非线性同余法、混沌法、平方取中法、组合法、小数开方法、Fibonacci 序列法等。这些方法或者直接递推地产生实值信号序列，或者先产生伪随机的整数或二进制序列，再通过除法等处理获得实值信号序列。以下介绍 3 个在信号处理中经常使用的方法。

(1) 乘线性同余法

该方法的信号序列产生式为

$$\begin{cases} x_i = ax_{i-1}(\text{mod } m) & x_i,m\in\mathbf{Z} \\ r_i = \dfrac{x_i}{m} & r_i\in\mathbf{R} \end{cases} \tag{14.93}$$

其中，**R** 表示实数域；r_i 表示输出的实数信号序列。关于乘线性同余法实数信号发生器有以下主要性质：①当 $m=2^L$、$L\geqslant 4$ 并且初值 x_0 为奇数时，取 $a\equiv 3(\mathrm{mod}\ 8)$ 或 $a\equiv 5(\mathrm{mod}\ 8)$，最大周期是 2^{L-2}；②当 m 为素数，取 a 与 m 互素，可以得到最大周期 $m-1$。

(2) 混合线性同余法

该方法的信号序列产生式与式(14.93)仅差一个加项：

$$\begin{cases} x_i = ax_{i-1}+c(\mathrm{mod}\ m) & x_i,m,c\in\mathbf{Z} \\ r_i = \dfrac{x_i}{m} & r_i\in\mathbf{R} \end{cases} \tag{14.94}$$

其中 $c>0$。为得到最大序列周期 m，算法需要满足：①c 与 m 互素；②对 m 的任何一个素因子 p，$a\equiv 1(\mathrm{mod}\ p)$；③若 4 是 m 的因子，则 $a\equiv 1(\mathrm{mod}\ 4)$。为满足上述条件，参数常取为

$$\begin{cases} m = 2^L & L\in\mathbf{Z} \\ a = 4\alpha+1 & \forall\alpha\in\mathbf{Z} \\ c = 2\beta+1 & \forall\beta\in\mathbf{Z} \\ x_0\geqslant 0 & \forall x_0\in\mathbf{Z} \end{cases} \tag{14.95}$$

(3) 混沌法

利用混沌映射可以产生伪随机的实值信号序列。Logistic 映射

$$x_{n+1} = ax_n(1-x_n) \quad a\in(0,4),\ x_n\in(0,1) \tag{14.96}$$

是常用的混沌映射之一，其中，x_n 为状态变量，a 为参数。当迭代次数 n 增加，数列 x_n 依选取的参数和初始状态存在 3 种变化可能：①收敛于不同点；②收敛于周期轨迹；③变为混沌。例如，当 $3.5\,699\,456\cdots\leqslant a\leqslant 4$，Logistic 映射将进入混沌状态。当 $x_0\in(0,1)$，Logistic 映射将产生一个在 $0\sim 0.25a$ 之间的信号，因此取 $a=4$，就得到 $0\sim 1$ 之间的信号。但是，由于序列是以概率密度

$$\rho(x) = \frac{1}{\pi\sqrt{x(1-x)}} \tag{14.97}$$

在 $0\sim 1$ 之间分布，为了在 $0\sim 1$ 内产生均匀分布的信号序列，可以求得分布函数

$$y(x) = \int_0^x \frac{1}{\pi\sqrt{x(1-x)}}\mathrm{d}x = \frac{2}{\pi}\arcsin\sqrt{x} \tag{14.98}$$

则根据定理 14.3.1，不难看出 $y(x)$ 是分布在 $0\sim 1$ 内的均匀分布信号序列。

混沌信号具有初值敏感性、难预测性、类似噪声的宽频特性、各态遍历性等统计性质，但理论周期仍然较难估计，一般可通过一些经验公式计算。

以下定理说明，可以利用统计近似抽样法将在 $0\sim 1$ 内均匀分布的信号序列变换为正态分布的。

定理 14.3.2　设 ξ_i 均是(0,1)上均匀分布的随机信号，$i=1,2,\cdots,L$，$\eta\sim N(\mu_\eta,\sigma_\mu^2)$ 是要获得的随机变量分布，当 $L\rightarrow\infty$ 时，有

$$\eta = \mu_\eta + \sigma_\eta\frac{\sum\limits_{i=1}^{N}\xi_i - \dfrac{L}{2}}{\sqrt{L/12}} \tag{14.99}$$

证明：由于

$$\mu_\xi = E(\xi_i) = \int_0^1 \xi_i \Pr(\xi_i) d\xi_i = \frac{1}{2} \tag{14.100}$$

$$\sigma_\xi^2 = \mathrm{Var}(\xi_i) = \int_0^1 (\xi_i - \mu_\xi)^2 \Pr(\xi_i) d\xi_i = \frac{1}{12} \tag{14.101}$$

根据中心极限定理，当 $L \to \infty$ 时，有

$$x = \frac{\sum_{i=1}^{L} \xi_i - L\mu_\xi}{\sqrt{L\sigma_\xi^2}} = \frac{\sum_{i=1}^{L} \xi_i - \frac{L}{2}}{\sqrt{L/12}} \sim N(0,1) \tag{14.102}$$

其中 $N(0,1)$表示均值为 0、方差为 1 的正态分布；对要获得的随机变量 $\eta \sim N(\mu_\eta, \sigma_\mu^2)$，有

$$\frac{\eta - \mu_\eta}{\sqrt{\sigma_\eta^2}} \sim N(0,1) \tag{14.103}$$

根据式(14.101)与式(14.102)，可令

$$\frac{\eta - \mu_\eta}{\sqrt{\sigma_\eta^2}} = \frac{\sum_{i=1}^{L} \xi_i - \frac{L}{2}}{\sqrt{L/12}}$$

即 $\eta = \mu_\eta + \sigma_\eta \dfrac{\sum_{i=1}^{N} \xi_i - \frac{L}{2}}{\sqrt{L/12}}$

试验验证，$N=12$ 时 η 的统计特性已经比较好，因此在应用中常用

$$\eta = \mu_\eta + \sigma_\eta \left(\sum_{i=1}^{12} \xi_i - 6 \right) \tag{14.104}$$

14.3.2 位平面替换与翻转

对用有限字长存储的数字信号或其变换系数，每个采样值均可看作整数，它们的低位比特对信号的感知效果影响很小。最简单的信息隐藏方法将每个样点的最低意义比特(LSB)替换为隐藏数据，若 $x(n)$表示载体信号，$w(n)$表示隐藏信号的比特串，这可以表示为

$$x'(n) = \begin{cases} x(n) + w(n) & x(n) \equiv 0 \pmod 2 \\ x(n) + w(n) - 1 & x(n) \equiv 1 \pmod 2 \end{cases} \tag{14.105}$$

以上方法在隐写和脆弱水印中得到了广泛应用。但一些应用要求载体数据可以被精确地复原，这使很多隐藏方法必须无损地压缩嵌入位置上的原数据，再将压缩数据和隐藏数据一并嵌入。由于 LSB 随机性较强，因此压缩效果不能满足要求，这样一些基于 LSB 的方法需要在更高位的比特上嵌入数据[6]。

定义 14.3.1 对用 N 比特长字节存储的数字信号采样点，它的全部 LSB 被称为 LSB 位平面，全部距离 LSB 最近的比特组成第 2 LSB 位平面，以下依次是第 3，4，…，N LSB 位平面。

与位平面相关的操作将引起一些信号特性的变化，因此这类操作还可以通过信号性质隐藏信息，而信息的提取可以通过检测信号的相关性质确定。位平面翻转是

主要的这类操作之一，它指将位平面上的置反，例如，在 LSB 和第 2 位平面上的翻转操作可分别记为

$$
\begin{aligned}
&F_{\text{LSB}}:\ 0\leftrightarrow 1,2\leftrightarrow 3,\cdots,254\leftrightarrow 255\\
&F_{\text{2ndLSB}}:\ 0\leftrightarrow 2,1\leftrightarrow 3,\cdots,254\leftrightarrow 252
\end{aligned}
\tag{14.106}
$$

可以更灵活地定义各种翻转操作，它们对不同的量化值进行不同的翻转。为了确保操作不过多地降低信号的感知质量，一般需要将以下定义的翻转强度控制在 6 以下。

定义 14.3.2　若 $P=\{0,\ 1,\cdots,\ 2^N-1\}$ 为信号 $x(n)$ 样点的量化级，则

$$
\frac{1}{|P|}\sum_{i,x(i)\in P}|x(i)-F(x)| \tag{14.107}
$$

被定义为翻转操作 F 对 $x(n)$ 的翻转强度，其中 $|P|$ 表示 P 中元素数量。

以下以翻转操作对局部信号 R-S(Regularity-Singularity)属性的改变说明相关的信号(属性)的调制原理。

定义 14.3.3　设将信号按每 N 个样点划分为样点分组，$F(\cdot)$ 是对信号分组的操作，$f(\cdot)$ 是以信号分组为输入的函数，对一个信号分组 G，可以在 $F(\cdot)$ 与 $f(\cdot)$ 下定义以下组别：

(1) 常规组 $\mathbf{R}$，$G\in\mathbf{R}$ 当且仅当 $f(F(G))>f(G)$；

(2) 奇异组 $\mathbf{S}$，$G\in\mathbf{S}$ 当且仅当 $f(F(G))<f(G)$；

(3) 不用组 $\mathbf{U}$，$G\in\mathbf{U}$ 当且仅当 $f(F(G))=f(G)$。

其中，分组的常规性质和奇异性质被统称为 R-S 属性。

以上定义说明可以通过翻转操作改变信号分组的 R-S 属性。在定义中，$f(\cdot)$ 常被称为区分函数，它需要反映操作 $F(G)$ 给 G 带来的变化。例如，当在应用中可以取

$$
f(x_1,x_2,\cdots,x_n)=\sum_{i=1}^{n-1}|x_{i+1}-x_i| \tag{14.108}
$$

极少出现 $G\in\mathbf{U}$ 的情况。根据翻转的性质，显然 $F(F(G))=G$，因此有

$$
G\in\mathbf{R}\Leftrightarrow F(G)\in\mathbf{S},\quad G\in\mathbf{S}\Leftrightarrow F(G)\in\mathbf{R},\quad G\in\mathbf{U}\Leftrightarrow F(G)\in\mathbf{U} \tag{14.109}
$$

这说明通过 $F(\cdot)$ 可以改变信号分组的 R-S 属性，若这些属性代表隐含的信息，则可以实现信息隐藏的功能，而通过检测这些属性可以提取信息。

需指出，在包含隐藏信息的数字内容未遭受改动时，具有 $\mathbf{U}$ 属性的分组能够在正常的检测中被识别，以上信息隐藏可以正确地提取全部嵌入数据，但在遭受改动时，相应被反替换和反翻转位置上的提取会出错，因此它们主要面向隐写和脆弱水印的应用。

14.3.3　扩频调制

扩频通信是一类将信号调制到各个频率上进行通信的方法。扩频通信有反电波截获的能力，这说明它本身有保护信息的作用，并且在移动通信中它具有码分复用(CDMA)的功能，这说明扩频信号可以在叠加传输的众多信号中被提取出来。以上特性使得 I. J. Cox 等人提出的扩频调制方法[7,8]也成为信息隐藏的主要方法之一。

扩频通信分为直接序列扩频(DSSS)和跳频扩频两类,信息隐藏中使用的扩频调制是基于前者的(见图14.3),它用伪随机序列将信息编码调制到各个频率分量上。若当前状态需隐藏的信源编码信息为一个比特 m,则典型的扩频调制和嵌入过程如下。

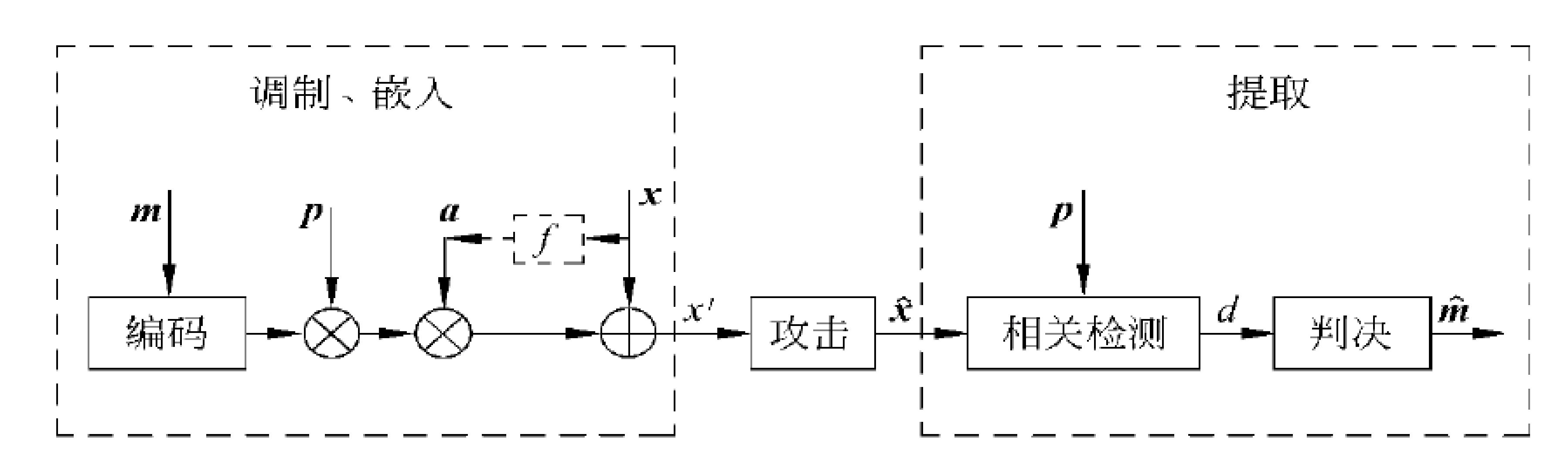


图 14.3 扩频调制、嵌入和提取流程

($\boldsymbol{m}$:水印消息;$\boldsymbol{p}$:伪随机序列;$\boldsymbol{a}$:调幅因子;$\boldsymbol{x}$:原始数据或其变换系数;$\boldsymbol{x}'$:发布数据;$\hat{\boldsymbol{x}}$:被攻击数据;d:相关值;$\hat{\boldsymbol{m}}$:提取水印;f:自适应函数)

(1) 比特扩展。$\boldsymbol{m}\leftarrow[mm\cdots m]_{L\times 1}$,其中,常数 L 被称为片率(chip rate),$\leftarrow$表示赋值。

(2) 扩频调制。$\boldsymbol{c}\leftarrow\boldsymbol{m}\otimes\boldsymbol{p}=(m\times p(1))\cdots(m\times p(L))$,其中,$\boldsymbol{p}$ 为伪随机信号序列,$\otimes$在这里表示逐样点相乘。

(3) 幅度调制。在幅调因子 $\boldsymbol{a}$ 与载体 $\boldsymbol{x}$ 无关时为 $\boldsymbol{s}\leftarrow\boldsymbol{a}\otimes\boldsymbol{c}=(m\times c(1))\cdots(m\times c(L))$,在相关时为 $\boldsymbol{s}\leftarrow\boldsymbol{a}\otimes\boldsymbol{c}=f(\boldsymbol{x})\otimes\boldsymbol{c}$。

(4) 嵌入。$\boldsymbol{x}'\leftarrow\boldsymbol{x}+\boldsymbol{s}$。

以上处理可综合地表示为

$$\boldsymbol{x}' := \boldsymbol{x}+\boldsymbol{s} = \boldsymbol{x}+\boldsymbol{a}\otimes\boldsymbol{m}\otimes\boldsymbol{p} \tag{14.110}$$

通过计算 $\boldsymbol{p}$ 与 $\boldsymbol{x}'$之间的相关性可以提取水印。若其大于选定的阈值 T,则认为嵌入的是1,否则是0。常用的相关性计算方法包括以下几种。

(1) 线性相关

$$C(\boldsymbol{x}',\boldsymbol{p}) = \frac{1}{L}\sum_{n=1}^{L}x'(n)p(n) = \frac{1}{L}<\boldsymbol{x}',\boldsymbol{p}> \tag{14.111}$$

(2) 归一化相关

$$\begin{aligned} Z(\boldsymbol{x}',\boldsymbol{p}) &= \left\langle \frac{\boldsymbol{x}'}{\sqrt{<\boldsymbol{x}',\boldsymbol{x}'>}}, \frac{\boldsymbol{p}}{\sqrt{<\boldsymbol{p},\boldsymbol{p}>}} \right\rangle \\ &= \frac{<\boldsymbol{x}',\boldsymbol{p}>}{\sqrt{<\boldsymbol{x}',\boldsymbol{p}>}\sqrt{<\boldsymbol{x}',\boldsymbol{p}>}} \end{aligned} \tag{14.112}$$

(3) 相关系数

$$\rho(\boldsymbol{x}',\boldsymbol{p}) = \frac{<\boldsymbol{x}'-E(\boldsymbol{x}'),\boldsymbol{p}-E(\boldsymbol{p})>}{\sqrt{<\boldsymbol{x}'-E(\boldsymbol{x}'),\boldsymbol{p}-E(\boldsymbol{p})>}\sqrt{<\boldsymbol{x}'-E(\boldsymbol{x}'),\boldsymbol{p}-E(\boldsymbol{p})>}} \tag{14.113}$$

在以上提取中,可能出现一定的漏报率和误报率,它们的定义分别是

$$e_{0,1} = \Pr(m=0 \mid m=1), \quad e_{1,0} = \Pr(m=1 \mid m=0) \tag{14.114}$$

设当 $m=1$ 时,相关值的均值和方差分别为 η_1 与 σ_1,当 $m=0$ 时分别为 η_0 与 σ_0,统计

结果表明,这些相关值通常满足正态分布,因此有

$$e_{0,1}=\int_{-\infty}^{T}\frac{1}{\sqrt{2\pi}\sigma_1}\mathrm{e}^{-\frac{(x-\mu_1)^2}{2\sigma_1^2}}\mathrm{d}x \tag{14.115}$$

$$e_{1,0}=\int_{T}^{\infty}\frac{1}{\sqrt{2\pi}\sigma_0}\mathrm{e}^{-\frac{(x-\mu_0)^2}{2\sigma_0^2}}\mathrm{d}x \tag{14.116}$$

以上信息隐藏方法被普遍地应用于鲁棒水印和隐写中。在鲁棒水印的应用中,一般需要考虑在水印攻击下的漏报情况,14.4.1 小节将给出一个 DCT 扩频水印的实例以及典型的攻击实验结果。

14.3.4　量化索引调制

基于量化和矢量量化技术,B. Chen 和 G. W. Wornell 提出了量化索引调制(QIM)水印方案,随后还提出了使用抖动(dither)调制(DM)和失真补偿(DC)的两种提高方案[9],以下分别称它们为 DM-QIM 与 DC-QIM 方案。本文以下将说明,基于 QIM 的方案能够消除载体信号对提取算法的干扰,因此也是较多使用的一类信息隐藏方法。

量化索引调制(QIM)通过选择不同的量化步长嵌入对应的信息编码比特,这可以用格(lattice)和陪集的概念描述。格是加群,它的元素是集合中的离散点,若集合为 L 维欧氏空间,这些点在矢量加法下组成了 L 维格。设 Λ' 为 Λ 的子格,对任意 $a\in\Lambda$,$a+\Lambda'$ 为 Λ' 的陪集,Λ' 的全体陪集是 Λ 的划分。设 QIM 水印方案的量化步长为 Δ,x 为原宿主信号嵌入域中的一个样点,$\mathbf{Z}$ 为整数集,则 ΔZ 是 $\pm\Delta/2$ 所生成格的子格,$\Lambda_0=\Delta Z$ 和 $\Lambda_1=\Delta/2+\Delta Z$ 是它的陪集,嵌入水印比特 $m\in\{0,1\}$ 可表示为

$$y=Q_m(x)=\arg_\lambda\min(|\lambda-x|),\quad \lambda\in\Lambda_m \tag{14.117}$$

Λ_0 和 Λ_1 也可以是相互间隔均匀的其他陪集。若考虑矢量量化,以上嵌入可一般地表示为

$$\boldsymbol{y}=Q_m(\boldsymbol{x})=\arg_\lambda\min(\|\boldsymbol{\lambda}-\boldsymbol{x}\|),\quad \boldsymbol{\lambda}\in\Lambda_m,m\in\{0,1\} \tag{14.118}$$

其中,$\boldsymbol{x}$、$\boldsymbol{y}$ 与 λ 为 L 维矢量;$\|\cdot\|$ 为向量的欧氏距离,Δ_i 为第 i 维的量化步长,$\Lambda_0=[\Delta_1 Z,\cdots,\Delta_L Z]$ 和 $\Lambda_1=[\Delta_1/2,\cdots,\Delta_L/2]+[\Delta_1 Z,\cdots,\Delta_L Z]$ 是 L 维子格 $[\Delta_1 Z,\cdots,\Delta_L Z]$ 的陪集,二维的情况参见图 14.4,该子格每个元素的第 i 个分量是 $\Delta_i Z$ 中的值,Λ_0 和 Λ_1 的间距称为量化码距 $d_{\min}$,在上述情况下它是

$$d_{\min}=\sqrt{\left(\frac{\Delta_1}{2}\right)^2+\cdots+\left(\frac{\Delta_L}{2}\right)^2} \tag{14.119}$$

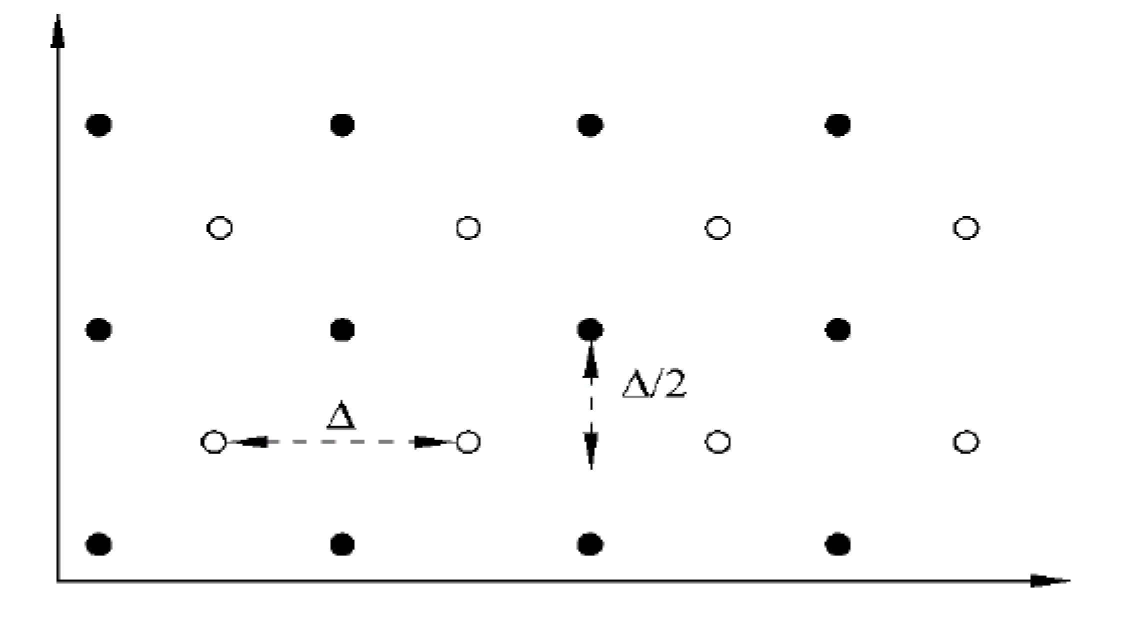


图 14.4　QIM 调制取值示意(实心点属于 Λ_0,空心点属于 Λ_1)

设 $\hat{\boldsymbol{y}}$ 为获得的发布版本,与上述嵌入对应的隐藏信息提取可以一般地表示为

$$\hat{\boldsymbol{m}}=\arg_{m\in\{0,1\}}\min(\|\hat{\boldsymbol{y}}-\Lambda_m\|) \tag{14.120}$$

抖动是提高嵌入信息隐蔽性的方法，DM-QIM 水印方案利用它在量化中嵌入水印。在量化步长为 Δ 的标量量化情况下，水印嵌入可表示为

$$y = Q_m(x) = Q(x - d_m) + d_m \quad m \in \{0,1\}, \tag{14.121}$$

设 Round(·)表示取最接近的整数，对信号样点 s，$Q(s)=\Delta \cdot \mathrm{Round}(s/\Delta)$，抖动值 d_0 和 d_1 是任何满足 $|d_0-d_1|=\Delta/2$ 的实数，如 $d_0=-\Delta/4$ 和 $d_1=\Delta/4$，有 $\Lambda_0=-\Delta/4+\Delta Z$，$\Lambda_1=\Delta/4+\Delta Z$。若采取矢量量化，式(14.121)中的变量需用矢量表示，其中，$Q(\boldsymbol{s})=[\Delta_1 \mathrm{Round}(s_1/\Delta_1),\cdots,\Delta_L \mathrm{Round}(s_L/\Delta_L)]$，$\boldsymbol{d}_0=[-\Delta_1/4,\cdots,-\Delta_L/4]^{\mathrm{T}}$ 和 $\boldsymbol{d}_1=[\Delta_1/4,\cdots,\Delta_L/4]^{\mathrm{T}}$ 为抖动向量，T 表示转置，Δ_i 可每次变化，$\Lambda_0=\boldsymbol{d}_0+[\Delta_1 Z,\cdots,\Delta_L Z]^{\mathrm{T}}$，$\Lambda_1=\boldsymbol{d}_1+[\Delta_1 Z,\cdots,\Delta_L Z]^{\mathrm{T}}$，全部抖动向量和量化步长形成了抖动序列和量化步长序列，它们可以作为密钥流使用。对应以上两种嵌入的水印检测都可用式(14.120)表示。DC-QIM 水印方案将量化误差乘上系数 $1-\alpha<1$ 后加回量化输出，可表示为

$$y = Q_m(\boldsymbol{x}) + (1-\alpha)(\boldsymbol{x} - Q_m(\boldsymbol{x})) \tag{14.122}$$

虽然加回的部分对水印检测也形成了干扰，但是它提高了感知质量，14.4.2 小节还将说明这也增加了基于 QIM 方法进行隐写的隐蔽性。

由于也存在码距的概念，基于 QIM 的调制方法类似于信道编码，因此可以借助相关的方法分析其解码的错误率[9]。设 m 为一个传输的比特，用方差为 σ_n 的高斯噪声模拟攻击，m' 为提取的比特，则

$$\Pr(m' \neq m) = \int_S^{\infty} \mathrm{e}^{-\frac{x^2}{2}} \mathrm{d}x, \quad S = \frac{d_{\min}}{2\sigma_n} \tag{14.123}$$

综合地看，若将水印嵌入和提取看作是水印信息的通信，QIM 类水印方案相对于 DSSS 方案的优势在于，宿主信号没有干扰通信，提取算法在一定的攻击强度内可以消除宿主信号的影响，实现了更大的码率。但是，14.4.2 小节将例示，由于载体信息被再量化的特征明显，基本的 QIM 不宜用于隐写。

14.3.5 统计量调制

信息隐藏也可以通过信号统计量的变化传递信息。本章前述的基于 R-S 属性和计算相关值的信息隐藏已说明，信号的特性往往被直接用于表征隐藏的信息。本小节将再给出两类典型的统计量调制方法。

1. 调制与基本数字特征相关的统计量

这类信息隐藏方法一般将载体信号分为成对的样点分组，并不同地调制它们，提取算法根据两个分组统计量之间的差值判断嵌入的数据或者是否有信息嵌入。若将该差值作为一个新的统计量 q，则算法仅当其大于一个阈值 T 时认为嵌入了值为 1 的比特或者有隐藏信息，其错误分析类似于式(14.115)与式(14.116)。

W. Bender 等人提出了分块调节信号均值的嵌入方法[10]。它随机选择两组信号样点 $A=\{a_i\}$ 与 $B=\{b_i\}$，它们均含有 n 个样点；为嵌入一个值为 1 的比特，将 A 与 B 中的每个样点分别加、减 d，使它们变为 $A=\{a'_i\}=\{a_i+d\}$ 和 $B=\{b'_i\}=\{b_i+d\}$，若用 μ_a、μ_b、$\mu_{a'}$ 与 $\mu_{b'}$ 分别表示 $\{a_i\}$、$\{b_i\}$、$\{a'_i\}$ 与 $\{b'_i\}$ 的均值，则 A 与 B 上样点均值的

差为

$$q=\mu_{a'}-\mu_{b'}=\mu_a+d-(\mu_b+d)=\mu_a-\mu_b+2d \tag{14.124}$$

由于统计上 μ_a 与 μ_b 接近，因此若有值为 1 的比特被嵌入，则 q 的值将明显增加。

有更多的信息隐藏方法选择了与均值、方差都相关的统计量进行调制和检测。N. Nikolaidis 与 I. Piva 提出的水印方案[11]也使用与前述类似的嵌入方法，但并不改动 B 中的样点，所选择的检验统计量为

$$q=\frac{\mu_{a'}-\mu_{b'}}{\sqrt{(\sigma_{a'}^2+\sigma_{b'}^2)/N}} \tag{14.125}$$

其中 $\sigma_{a'}$ 与 $\sigma_{b'}$ 分别表示 $\{a'_i\}$ 与 $\{b'_i\}$ 的均值。I. K. Yeo 与 H. J. Jim 提出的水印嵌入方案[12]也是基于 A、B 两个分组嵌入一个比特，它先计算 $S=\sqrt{(\sigma_a^2+\sigma_b^2)/(N-1)}$，再通过以下方法嵌入值为 1 的比特：

$$\begin{cases}a'_i=a_i+\operatorname{sgn}(\mu_a-\mu_b)\cdot d\cdot S/2\\ b'_i=b_i-\operatorname{sgn}(\mu_a-\mu_b)\cdot d\cdot S/2\end{cases} \tag{14.126}$$

其中 d 为常数。式(14.126)使 $\mu_{a'}$ 与 $\mu_{b'}$ 之差大于 CS，因此，根据统计量

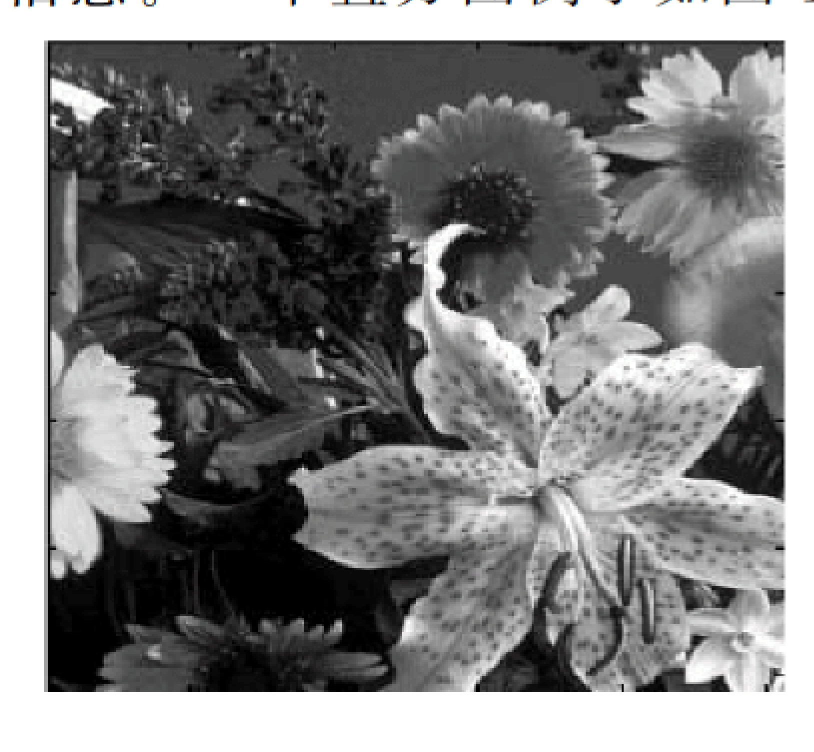
$$q=\frac{(\mu_{a'}-\mu_{b'})^2}{S^2} \tag{14.127}$$

是否大于阈值可以判断是否嵌入了值为 1 的比特。另外，基于调制与检测 A 与 B 中样点的能量差也可以嵌入、判断嵌入的信息。

2. 调制直方图

定义 14.3.4　若有 N 个样点的信号 $x(n)$ 有 L 个量化级 r_k，$k=0,1,\cdots,L-1$，若 n_k 表示等于量化级 r_k 的样点数量，则描绘 $P(r_k)=n_k$ 的曲线被称为直方图。若将 r_k 和 n_k 均做归一化处理，即 $r_k\leftarrow r_k/r_{L-1}$，$n_k\leftarrow n_k/N$，描绘 $P(r_k)=n_k$ 的曲线被称为归一化直方图。

在信息隐藏的各类应用中都有基于调制直方图的方案[13]。它们的基本方法是，将载体信号分为一些分块，将每个分块的直方图调节到特定的形状，即使样点值满足特定的分布函数(被称为规定化处理)，并约定不同形状的直方图表示不同的信息。一个直方图例子如图 14.5 所示。



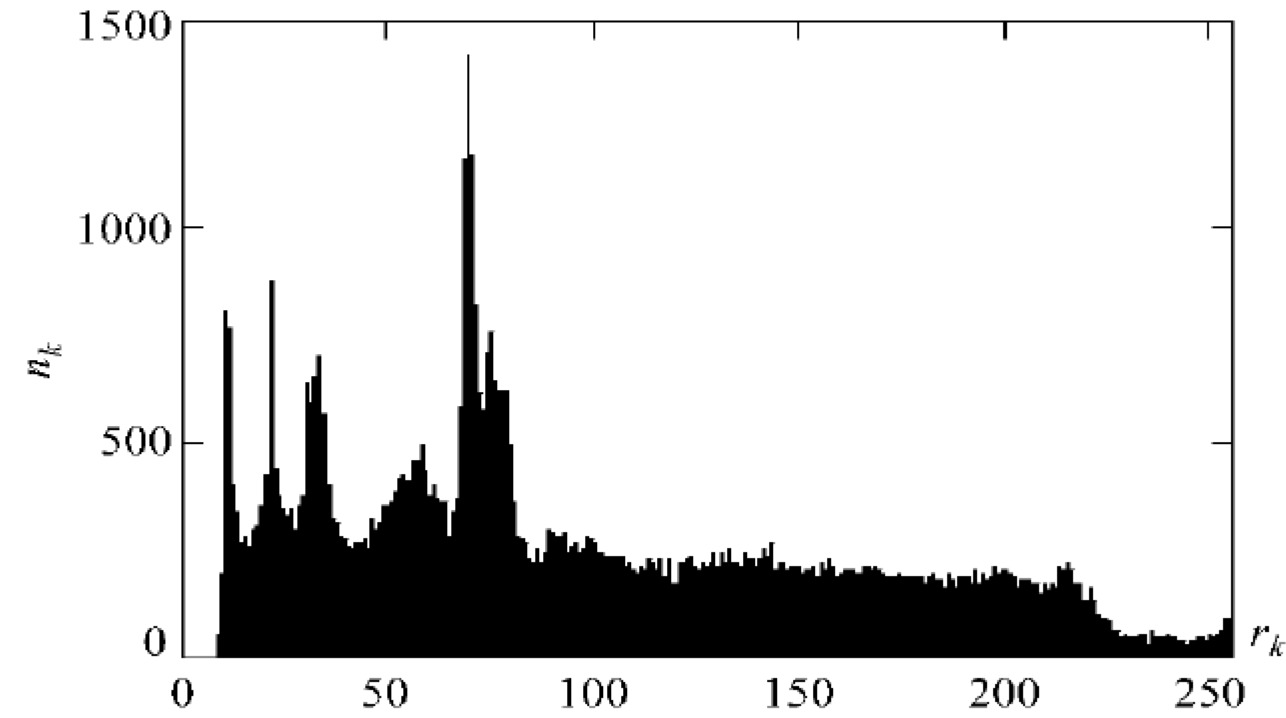


图 14.5　一个 256×256 灰度图像的直方图例子

在假设变量均连续的情况下，以下给出规范化处理归一化直方图的原理，根据它可以直接获得规范化直方图的方法。若量化级 x 满足 $0 \leqslant x \leqslant 1$，可以通过 $y=T(x)$ 将每个 x 对应到一个新的值，$T(x)$ 显然需要满足：①单调增加，这保证了量化级由低到高的次序不变；② $0 \leqslant T(x) \leqslant 1$。若用 X 与 Y 表示取值为 x 与 y 的随机变量，由概率论的基本定理可得以下定理。

定理 14.3.3 若 $Y=T(X)$，并且 $T(X)$ 在 X 的取值范围内单调增加，则

$$F_Y(y) = F_X(T^{-1}(y)) \tag{14.128}$$

其中 $F_Y(y)$ 与 $F_X(x)$ 分别为 X 与 Y 的分布函数。

证明：由于 $T(X)$ 单调增加，则以下两个事件是等价的

$$X \leqslant T^{-1}(y), \quad T(X) \leqslant y$$

则

$$\begin{aligned} F_Y(y) &= \Pr(Y \leqslant y) = \Pr(T(x) \leqslant y) \\ &= \Pr(X \leqslant T^{-1}(y)) = F_X(T^{-1}(y)) \end{aligned}$$

定理 14.3.4 设 $f_Y(y)$ 与 $f_X(x)$ 分别为以上 X 与 Y 的概率密度函数，若令

$$y = T(x) = \int_0^x f_X(\omega)\mathrm{d}\omega \tag{14.129}$$

则 $f_Y(y)=1$。

证明：根据式(14.129)，$T(x)$ 显然满足以上要求的单调增加和 $0 \leqslant T(x) \leqslant 1$ 两个性质。根据定理 14.3.3 可得

$$f_Y(y) = f_X(T^{-1}(y))\frac{\mathrm{d}}{\mathrm{d}y}T^{-1}(y) = f_X(x)\frac{\mathrm{d}}{\mathrm{d}y}x\,\Big|_{x=T^{-1}(y)} \tag{14.130}$$

对式(14.129)求导后有

$$\frac{\mathrm{d}y}{\mathrm{d}x} = f_X(x) \tag{14.131}$$

将式(14.131)代入式(14.130)中得到

$$f_Y(y) = f_X(x)\frac{\mathrm{d}}{\mathrm{d}y}x\,\Big|_{x=T^{-1}(y)} = \frac{\mathrm{d}x}{\mathrm{d}y}\frac{\mathrm{d}x}{\mathrm{d}y}\,\Big|_{x=T^{-1}(y)} = 1$$

式(14.129)的操作被称为使直方图均匀化，基于它可以得到将直方图规定化的方法。首先将原信息进行直方图均衡化：

$$y = T(x) = \int_0^x f_X(\omega)\mathrm{d}\omega \tag{14.132}$$

假定 z 是满足规定化要求的信号，它的概率密度为 $f_Z(z)$，则实际也存在以下均衡化操作：

$$u = G(z) = \int_0^z f_Z(\omega)\mathrm{d}\omega \tag{14.133}$$

则 $z=G^{-1}(u)$，由于 y 和 u 都是均衡化后的结果，它们有相同的概率密度，因此用 y 代替 u 不影响 z 的概率密度，则

$$z = G^{-1}(y) = G^{-1}(T(x)) \tag{14.134}$$

可以得到所需的 z。用样点出现的频度代替概率密度值，可以将以上过程离散化。

14.4　应用举例

通过介绍 DCT 域的扩频鲁棒数字水印方案和小波域的基于量化索引调制(QIM)的隐写方案，本节例示了前述主要信号处理技术的在信息隐藏中的基本使用方法。

14.4.1　DCT 域扩频鲁棒水印与攻击

鲁棒数字水印(以下简称水印)[1]技术是一种针对数字内容的版权保护技术，它通过在原内容的感知冗余数据中隐蔽地嵌入包含版权信息的数据，实现对数字内容的各类保护，这些隐藏的数据或信号被称为数字水印。水印技术有 3 个主要性质：感知透明性、鲁棒性和安全性。感知透明性指人类难以感知水印的存在，水印不影响数字内容的正常使用；鲁棒性指水印难以被攻击者除去，任何能够损害水印的操作也会造成数字内容的质量严重受损；安全性是指水印与其验证协议难以被低代价的攻击所破坏。

水印技术可按照嵌入位置分为时空域水印和变换域水印两类，它们也可按照 14.3 节介绍的嵌入方法分类。其中，DCT 域的扩频水印最早由 I. J. Cox 等人提出[7]并由 M. Barni 等人进一步完善[8]，是一种非常典型的变换域水印，它主要分为水印嵌入和水印检测两部分。以下以 2 维灰度图像作为原始信号分别描述其具体步骤。

1. 水印嵌入

(1) 生成水印序列。使用式(14.104)提供的方法生成长度为 M 的接近正态分布的序列 $\boldsymbol{X}=\{x_1,x_2,\cdots,x_M\}$ 作为水印，它可代表数字内容所有者的标识。

(2) DCT 变换。对 $N\times N$ 的原图像 $\boldsymbol{I}$ 进行二维 DCT 变换，并将变换系数矩阵 $\boldsymbol{D}$ 按照 zigzag 顺序排列为一维向量 $\boldsymbol{T}$，其中，zigzag 顺序可以用图 14.6 例示，它的排列特点是元素按照频率从低到高排列。

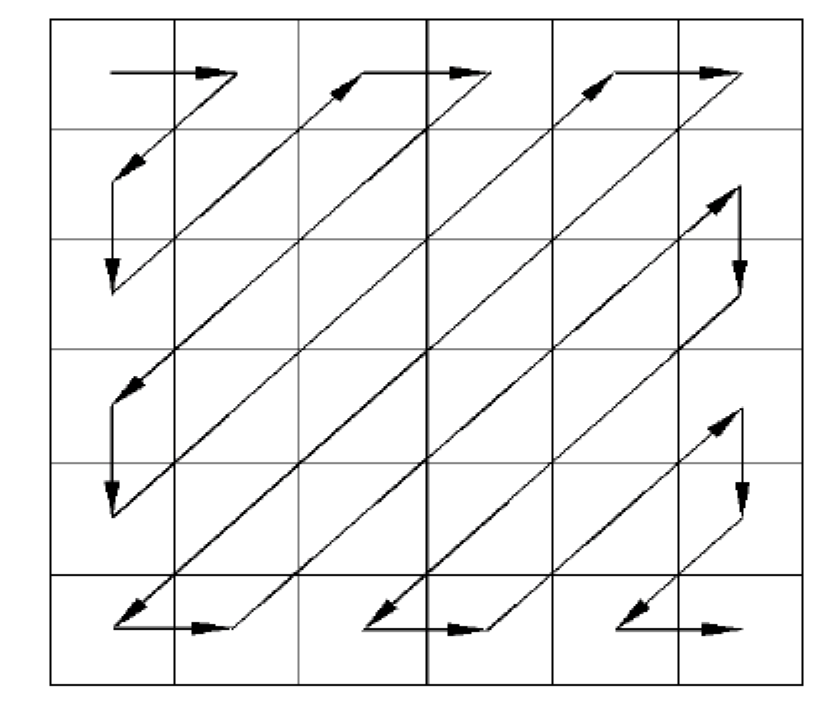


图 14.6　6×6 矩阵元素的 zigzag 顺序（每个方格代表一个元素）

(3) 嵌入水印。为了在保持鲁棒性的同时获得更好的感知透明性，算法将水印嵌入中频区域，因此取出向量 $\boldsymbol{T}$ 的第 $(L+1)$ 至 $(L+M)$ 个元素 $\{t_{L+1},t_{L+2},\cdots,t_{L+M}\}$ 作为嵌入位置，计算向量 $\boldsymbol{T}'=\{t'_{L+1},t'_{L+2},\cdots,t'_{L+M}\}$，其中

$$t'_{L+i}=t_{L+i}+\alpha\cdot|t_{L+i}|\cdot x_i,\quad i=1,2,\cdots,M \tag{14.135}$$

常数 α 用于调节嵌入强度；随后用 $\boldsymbol{T}'$ 替换 $\boldsymbol{T}$ 中相应位置上的系数，进行反 zigzag 排序，恢复 2 维系数矩阵并进行 DCT 反变换，得到嵌入水印后的图像 $\boldsymbol{I}'$。

2. 水印检测

对可能含有水印 $\boldsymbol{X}$ 的待测图像 $\boldsymbol{I}^*$ 进行 DCT 变换和 zigzag 排列，取第$(L+1)$至$(L+M)$个 DCT 系数构成向量 $\boldsymbol{T}^*=\{t_{L+1}^*,t_{L+2}^*,\cdots,t_{L+M}^*,\}$，计算向量 $\boldsymbol{T}^*$ 与 $\boldsymbol{X}$ 的线性相关值

$$z=\frac{<\boldsymbol{X},\boldsymbol{T}^*>}{M}=\frac{1}{M}\sum_{i=1}^{M}x_i t_{L+i}^* \tag{14.136}$$

作为判决依据：若 $\lg(z)$不小于阈值 T_z，则认为 $\boldsymbol{I}^*$ 中含有水印 $\boldsymbol{X}$，反之则没有。

以上检测方法存在误报和漏报的可能，对前者可以建立相关的统计模型分析，但由于需要考虑不同的攻击，对后者一般用实验分析。假设在 $\boldsymbol{I}^*$ 不含水印的情况下，z 近似服从正态分布 $N(\mu,\sigma^2)$，则误报率 P_f 为

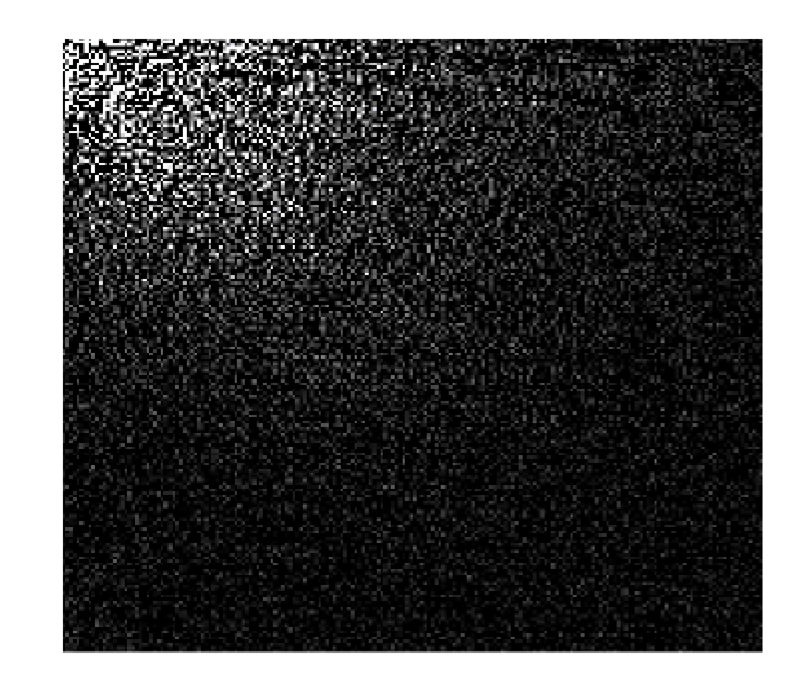
$$P_f=\frac{1}{\sqrt{2\pi\sigma^2}}\int_{T_z}^{\infty}\mathrm{e}^{-\frac{(x-\mu)^2}{2\sigma^2}}\mathrm{d}x \tag{14.137}$$

由于攻击者发现误报后可以谎称其他水印的存在，P_f 是衡量水印安全性的重要因素。M. Barni 等人经过估算发现[8]，当 $M=16\ 000$、$\alpha=0.01$、阈值 $T_z=0.034$ 时，以上算法的 P_f 约在10^{-6}量级。

图 14.7 给出了水印嵌入和检测的例子。其中，水印检测值 z 均取对数后显示。可以看出，嵌入水印后的图像与原始图像相比，在人眼的视觉感知上几乎没有差别，体现了算法的感知透明性；而图 14.7(d)所示说明当检测水印正确时，检测值将显著增加。



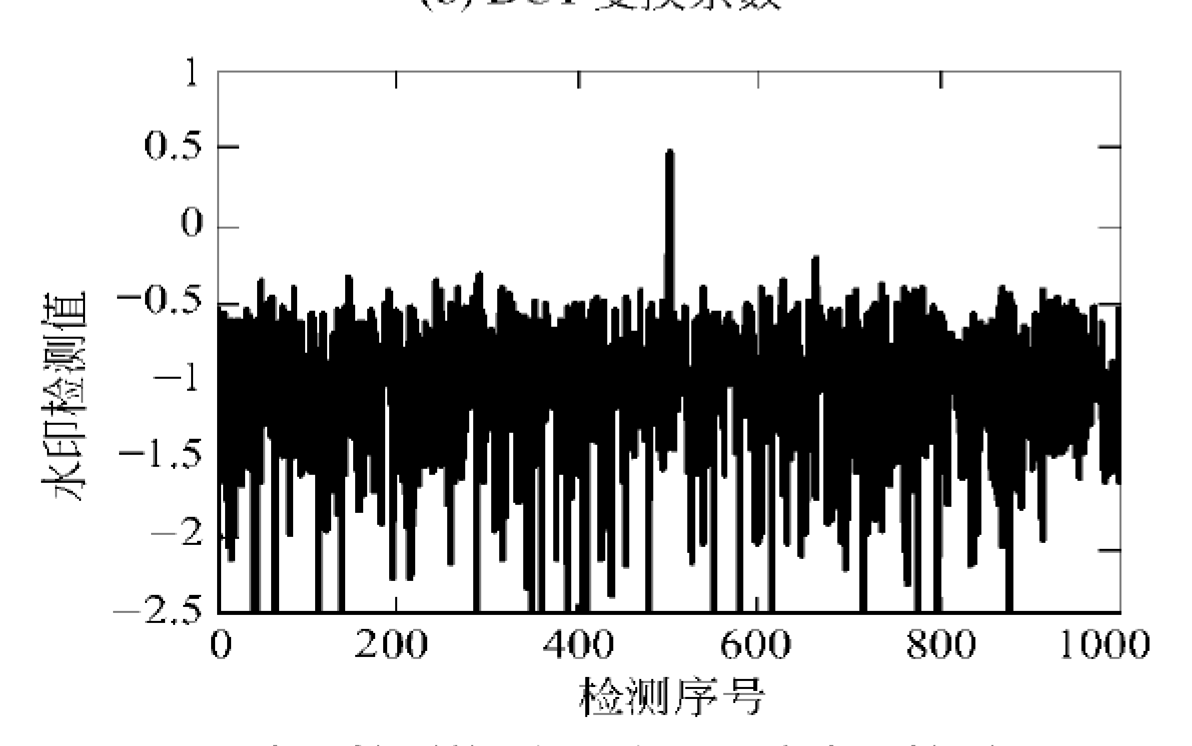
(a) 512×512 原始图像



(b) DCT 变换系数



(c) 嵌入水印后的图像 (PSNR=48.16dB)



(d) 水印检测值 (仅 1 次用正确水印检测)

图 14.7　DCT 域扩频水印嵌入与检测情况

对图像水印常见的攻击包括图像有损压缩、添加噪声、尺度缩放等。图 14.8 至图 14.10 依次给出了经过这 3 种攻击后水印的检测情况，其中具有较高检测值时均使用了正确的水印进行检测。可以看出，在经过一定程度的上述攻击后，水印的存在性仍能够正确地被识别，体现了 DCT 域扩频水印方法的鲁棒性。



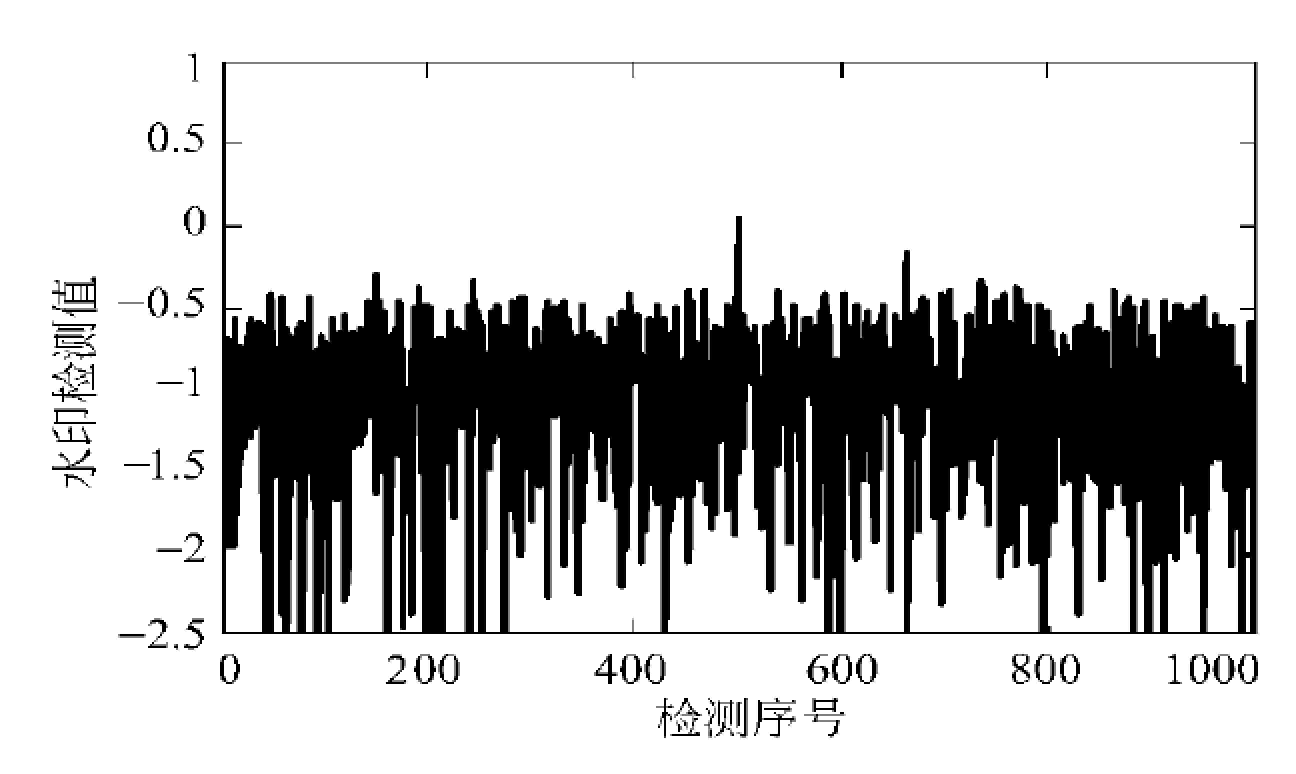


图 14.8　品质参数为 4%、平滑参数为 0 的 JPEG 压缩后的图像(左)及水印检测(右)



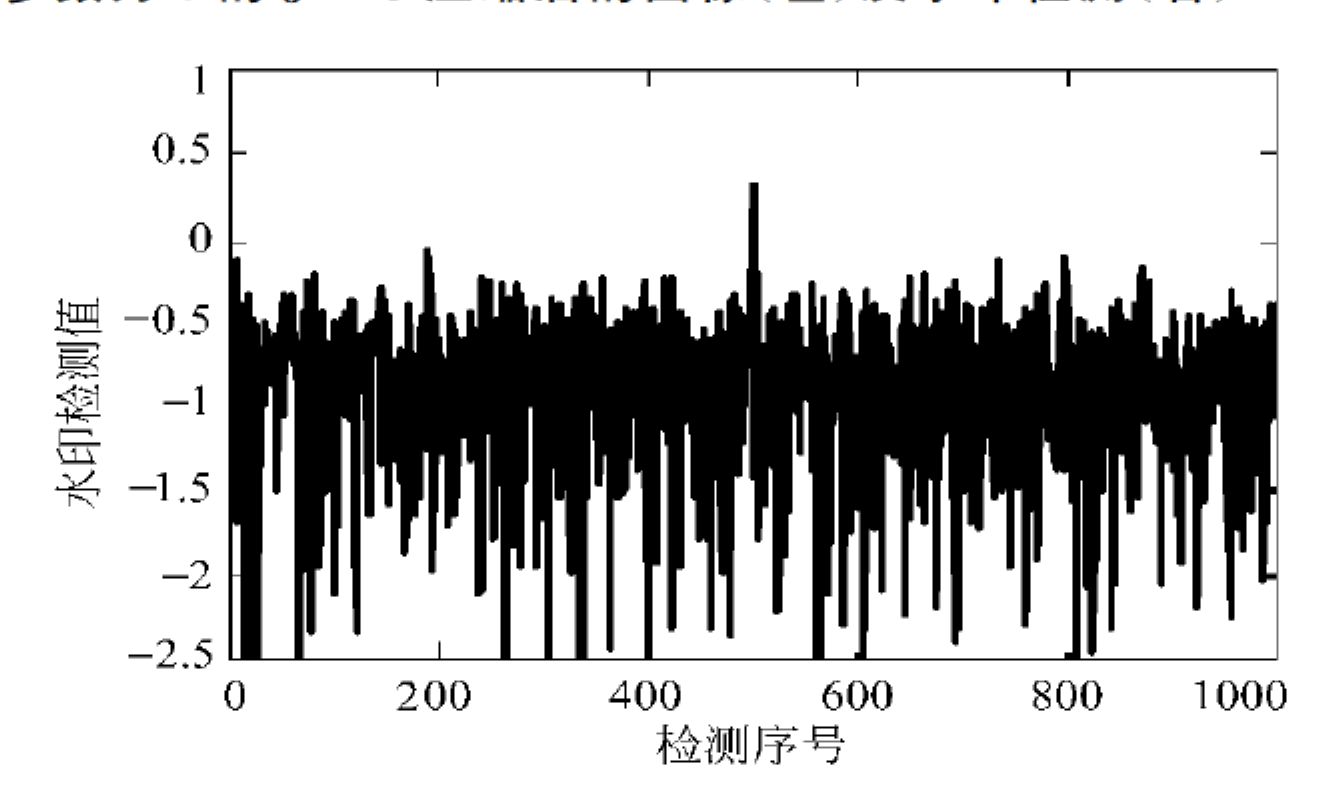


图 14.9　添加高斯白噪声后的图像(左，PSNR 值降至 22.18dB)及水印检测(右)



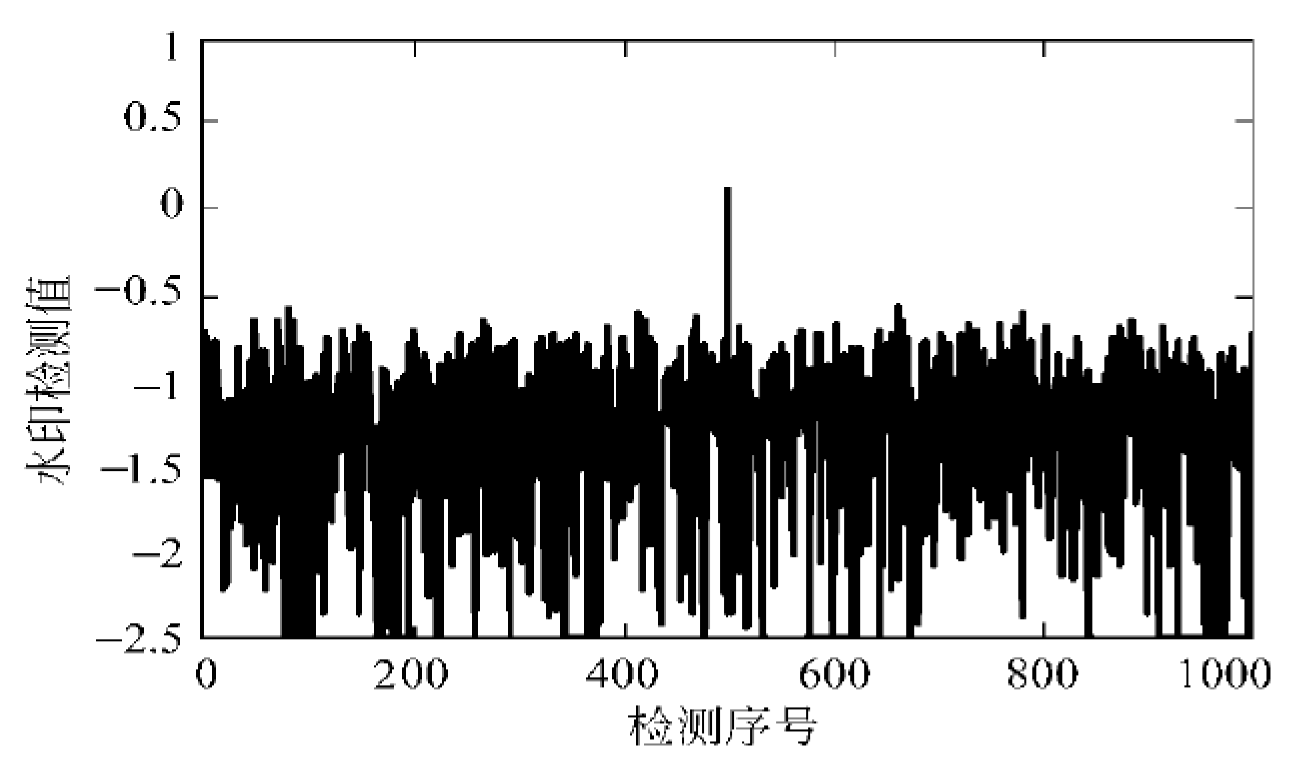


图 14.10　尺寸缩小为 256×256 后的图像(左)及水印检测(右)

14.4.2 小波域 QIM 隐写与分析

隐写[2]是信息隐藏的重要分支，它将机密信息（亦称隐秘信息）隐蔽地嵌入在载体内容中，通过载体的传输实现保密通信。隐蔽性是评价隐写是否安全的关键因素，即不能让第三方检测出隐写事实的存在。隐写按照信息嵌入域的不同，可以分为时空域方法和变换域方法两类，本小节将介绍一类在小波域利用 B. Chen 和 G. W. Wornell 提出的 QIM[9] 进行图像隐写方法，也将介绍对它们的典型分析方法。

这里首先介绍一种基本的图像 QIM 隐写。它的输入是载体图像 X 和隐秘信息 $\boldsymbol{m}$，嵌入算法包括以下步骤：

（1）小波多分辨率分解。用 14.2.3 小节给出的小波滤波器组（以下实验使用长度为 8 的 Daubechies 小波）对图像 X 进行小波分解，得到在 LL、LH、HL、HH 子带的分解系数 $C(x,y)$、$H(x,y)$、$V(x,y)$、$D(x,y)$；选择 LL 子带系数 $C(x,y)$ 作为隐秘信息嵌入位置。

（2）嵌入隐秘信息。为利用 QIM 将隐秘信息 $\boldsymbol{m}$ 嵌入到 $C(x,y)$ 中，定义量化函数

$$Q_i(x) = Q(x - d_i) + d_i \tag{14.138}$$

其中，$Q(x)=q\text{Round}(x/q)$，q 为量化步长，$d_0=-q/4$，$d_1=q/4$，则嵌入规则为

$$C'(x,y) = \begin{cases} Q_0(C(x,y)) & m(k) = 0 \\ Q_1(C(x,y)) & m(k) = 1 \end{cases} \tag{14.139}$$

其中 $m(k)$ 代表将嵌入在 $C(x,y)$ 位置上的隐秘信息比特。

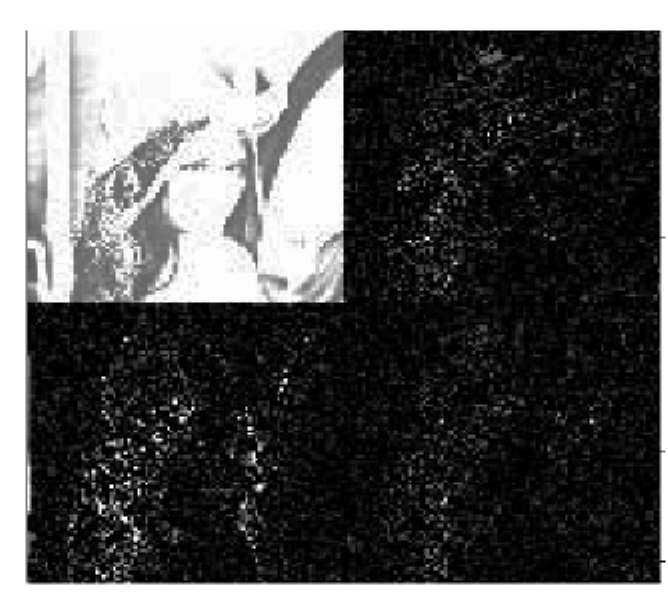
（3）图像重构。对量化后的系数进行小波多分辨率分解的综合，得到隐写图像 $\boldsymbol{X}'$。

隐写提取算法首先对 $\boldsymbol{X}'$ 进行上述小波分解，得到 LL 子带系数 $C''(x,y)$，再利用以下反向量化规则从 $C''(x,y)$ 中提取信息：计算 $p=\text{Round}((C''(x,y)-q/4)/(q/2))$，其中 $\text{Round}(x)$ 表示取最接近 x 的整数，则提取隐秘信息为

$$m'(k) = \begin{cases} 1 & p = 2n \\ 0 & p = 2n+1 \end{cases} \tag{14.140}$$

若取量化步长 $q=12$，以上算法可以在 512×512 的图像中嵌入了 256×256 比特的隐秘信息，图 14.11 所示为实验图像和小波分解系数。


(a) 512×512载体图像


(b) 小波分解系数


(c) 隐写图像 (PSNR=46.3)

图 14.11 一次小波域 QIM 隐写中的图像和变换系数

对 QIM 隐写最典型的攻击方法是直方图分析。由于所有样点均调制为量化值的倍数，QIM 隐写会减少嵌入域样点的数值个数，这种变化会反映在样点的直方图上，使得嵌入域的直方图会由相对连续的变为相对离散的，因此，通过直方图分析能够检测出载体图像是否含有隐秘信息。图 14.12 所示为载体图像和隐写图像在嵌入域的直方图，可以看出，载体图像在嵌入域的直方图更连续，而隐写图像在嵌入域的直方图更离散，并且峰值为载体图像直方图的数倍。

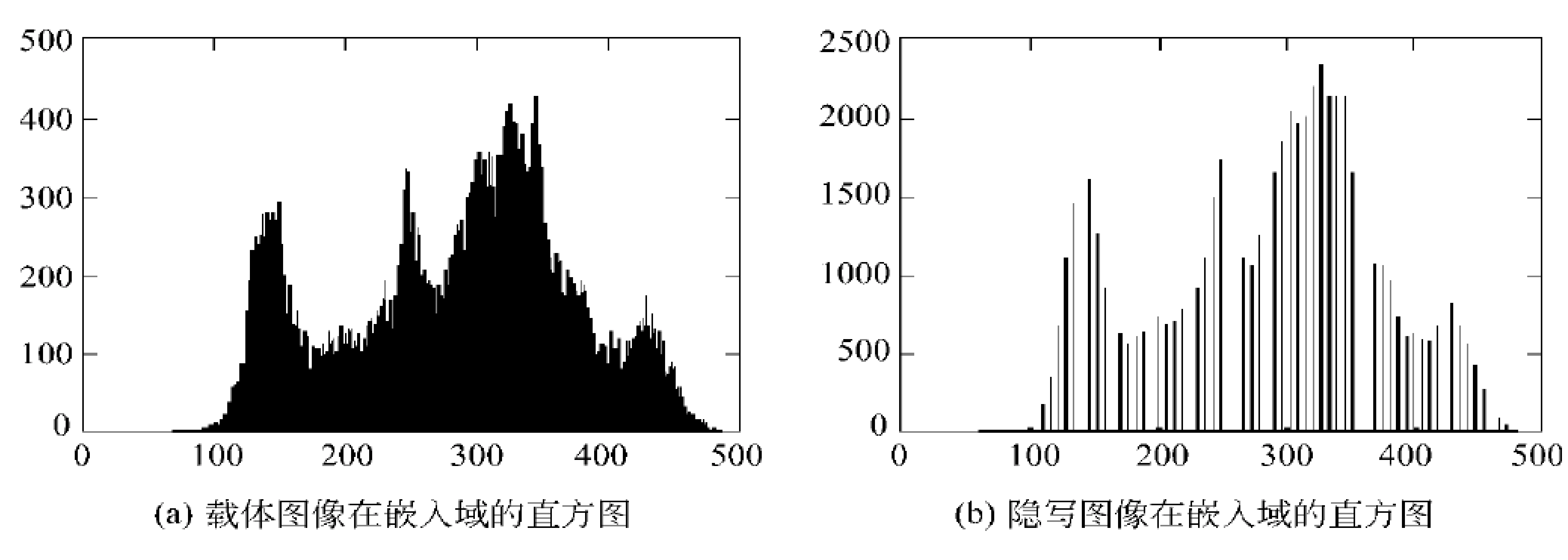


图 14.12　载体图像和使用基本 QIM 隐写后图像嵌入域的直方图

为加强隐写信息的隐蔽性，可以对上述基本 QIM 隐写方法进行改进，将以上步骤(2)使用的 QIM 替换为基于失真补偿(Distortion-Compensated)的 QIM，得到 DC-QIM 隐写方法。它仅将第一种方法的嵌入规则修改为

$$C'(x,y)=\begin{cases}Q_0(\alpha C(x,y))+(1-\alpha)C(x,y), & m(k)=0\\ Q_1(\alpha C(x,y))+(1-\alpha)C(x,y), & m(k)=1\end{cases}\tag{14.141}$$

其中 $0<\alpha<1$。提取算法的反向量化规则相应变为 $p=\mathrm{Round}((\alpha C''(x,y)-q/4)/(q/2))$，隐秘信息仍由式(14.140)提取。由于该 QIM 方法在量化强度和图像失真之间做了折衷，因此减弱了 QIM 的特征，提高了隐写的隐蔽性。实验取 $\alpha=0.6$，得到如图 14.13 所示的结果。可以看出，算法改进后，隐写图像在嵌入域的直方图更连续，但与载体图像的直方图仍有区别。

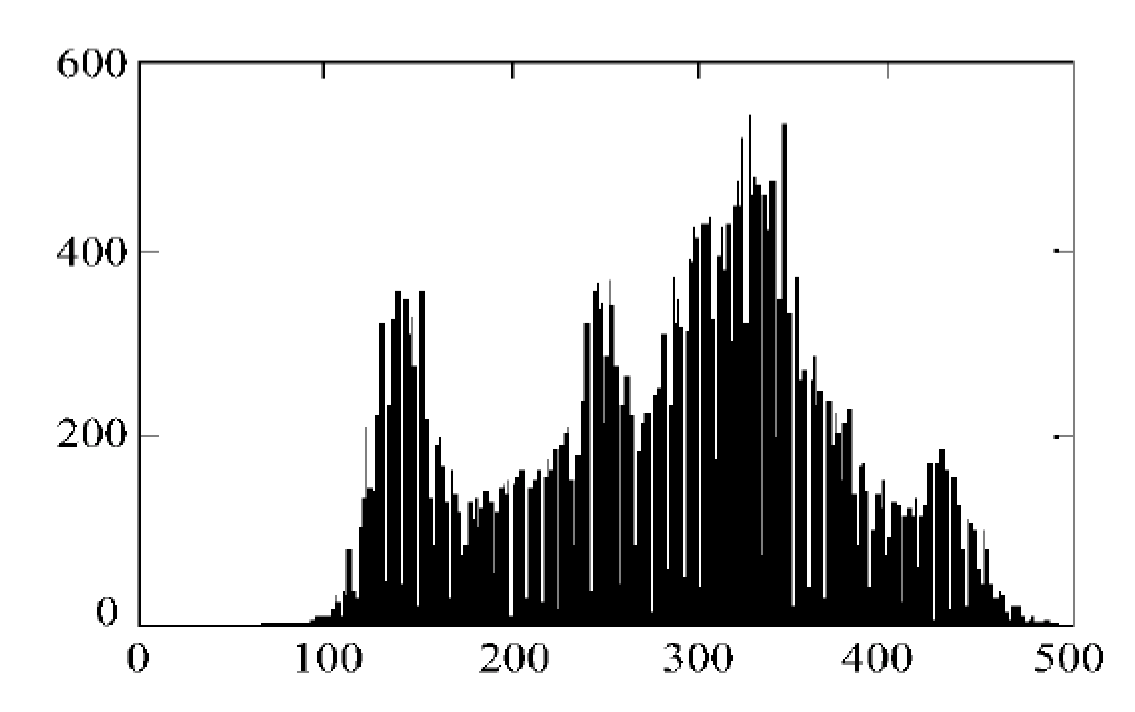


图 14.13　使用 DC-QIM 隐写后图像嵌入域的直方图

为进一步加强隐蔽性，可以将第一种隐写方法中的 QIM 替换为使用抖动调制(Dither Modulation)的 QIM，得到 DM-QIM 隐写方法。它的嵌入算法包括以下

步骤。

（1）由小波多分辨率分解得到 $C(x,y)$。

（2）将隐蔽信息 $\boldsymbol{m}$ 划分为长度为 L 比特的分块，记为 $\{\boldsymbol{m}_1,\boldsymbol{m}_2,\cdots,\boldsymbol{m}_{N/L}\}$，$N$ 为隐蔽信息的比特长度。

（3）在 $[q_{\min},q_{\max}]$ 范围内随机生成长度为 L 的序列 $q_1,q_2,\cdots,q_L$，其中，$q_{\min}>0$，$q_{\max}>0$；在 $[-q_k/2,q_k/2]$ 范围内随机生成 $d(k,0)$，$1\leqslant k\leqslant L$，并根据式(14.142)计算得 $d(k,1)$，即

$$d(k,1)=\begin{cases} d(k,0)+q_k/2, & d(k,0)\leqslant 0 \\ d(k,0)-q_k/2, & d(k,0)>0 \end{cases} \quad k=1,2,\cdots,L \tag{14.142}$$

$d(k,0)$ 和 $d(k,1)$ 称为长度为 L 的抖动序列，由此定义量化函数为

$$\begin{aligned} Q_i(x(k)) &= Q(x(k)-d(k,i))+d(k,i) \\ &= q_k \mathrm{Round}((x(k)-d(k,i))/q_k)+d(k,i) \end{aligned} \tag{14.143}$$

（4）将载体图像小波分解系数 $C(x,y)$ 同样划分为长度为 L 的分块，记为 $\{\boldsymbol{c}_1,\boldsymbol{c}_2,\cdots,\boldsymbol{c}_{N/L}\}$，分别在它的第 j 个分块 $\boldsymbol{c}_j$ 嵌入隐蔽信息 $\boldsymbol{m}_j$，嵌入规则为

$$c'_j(k)=\begin{cases} Q_0(c_j(k)), & m_j(k)=0 \\ Q_1(c_j(k)), & m_j(k)=1 \end{cases} \quad k=1,\cdots,L;j=1,\cdots,N/L \tag{14.144}$$

将各分块的量化后系数 $\boldsymbol{c}'_j$ 组合为 $C'(x,y)$。

（5）对 $C'(x,y)$ 进行小波多分辨率分解重构，得到隐写图像 $\boldsymbol{X}'$。

DM-QIM 提取算法首先对 $\boldsymbol{X}'$ 进行上述小波分解，得到 LL 子带系数 $C''(x,y)$，再将 $C''(x,y)$ 划分为长度为 L 的分块，记为 $\{\boldsymbol{c}''_1,\boldsymbol{c}''_2,\cdots,\boldsymbol{c}''_{N/L}\}$，对每一个分块分别计算 $p_j(k)=\mathrm{Round}((c''_j(k)-d(k,0))/(q_k/2))$，则隐蔽信息比特的提取为

$$m'_j(k)=\begin{cases} 0, & p_j(k)=2n \\ 1, & p_j(k)=2n+1 \end{cases} \quad k=1,2,\cdots,L;j=1,2,\cdots,N/L \tag{14.145}$$

最后，将得到的 $\{\boldsymbol{m}'_1,\boldsymbol{m}'_2,\cdots,\boldsymbol{m}'_{N/L}\}$ 组合为隐蔽信息 $\boldsymbol{m}'$。

DM-QIM 隐写对量化步长和抖动序列都做了随机处理，进一步减弱了 QIM 的特征。当实验取 $L=1024$、q_k 为在 $[10,20]$ 范围内随机生成的整数时，从实验结果(见图 14.14)可以看出 QIM 直方图的量化特征进一步减弱。若每个分块上的量化

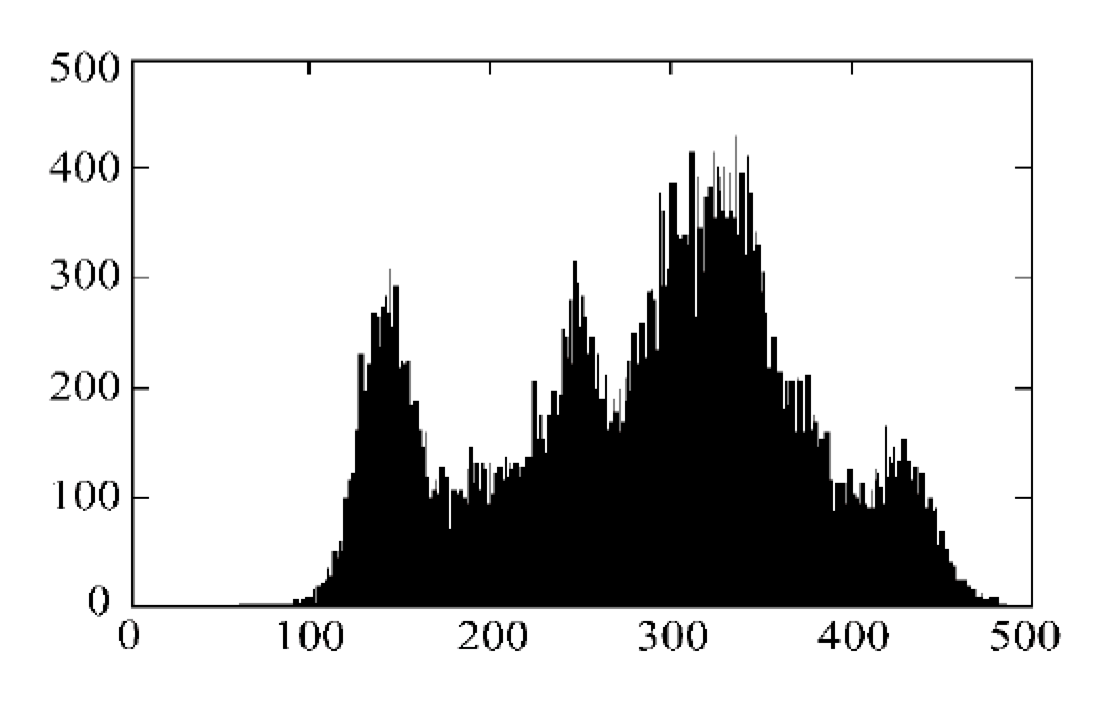


图 14.14　使用 DM-QIM 隐写后图像嵌入域的直方图

步长和抖动序列都不同，则隐蔽性还可以进一步加强。

14.5　注记

本章介绍了在信息隐藏中常用的一些数字信号处理技术，并用典型的数字水印和隐写实例给出了这些技术在信息隐藏中的应用方法。数字信号处理是一门得到广泛研究和应用的领域，相关研究成果和应用实例很多，并且仍然在不断涌现，其中很多成果均可运用到现代信息隐藏技术中去。在数字信号处理方面出现了很多好的教材，如文献[3]～[5]，它们对获得更好的信息隐藏及其分析方法都有帮助作用。近年来，也出现了一些信息隐藏专著，如文献[1]和[2]，它们更系统地阐述了信息隐藏的各类方法。另外，有关信息隐藏的最新进展可以在《IEEE Transactions on Image Processing》、《IEEE Transactions on Multimedia》、《IEEE Transactions on Information Forensics and Security》、《IET Information Security》、《Signal Processing》等期刊以及《International Workshop on Information Hiding》(IH)、《International Workshop on Digital Watermarking》(IWDW)等会议文集中找到。

参考文献

[1] Cox I J, Miller M L, Bloom J A. Digital Watermarking, Burlington, MA: Morgan Kaufmann Publishers, Elsevier Science (USA), 2002

[2] Katzenbeisser S, Peticolas F A P. Information Hiding Techniques for Steganography and Digital Watermarking, Norwood, MA: Artech House, Inc., 2000

[3] Oppenheim A V, Schafer R W, Buck J. R. Discrete-Time Signal Processing. Englewood Cliffs, NJ: Prentice-Hall, Inc., 1999

[4] 丁玉美，阔永红，高新波. 数字信号处理——时域离散随机信号处理. 西安：西安电子科技大学出版社，2002

[5] Burrus C S, Gopinath R A, Guo H. Introduction to Wavelet and Wavelet Transforms: A Primer. Englewood Cliffs, NJ: Prentice Hall, 1998

[6] Goljan M, Fridrich J, Du R. Distortion-free data embedding for images, In Proc. of IH 2001, LNCS, Vol. 2137, 27-41, Berlin, Germany: Springer-Verlag, 2001

[7] Cox, I J, Kilian J, Leighton F T, Shamoon T. Secure spread spectrum watermarking for multimedia, IEEE Transactions on Image Processing, Vol. 6, No. 12, 1673-1687, 1997

[8] Barni M, Bartolini F, Capellini V, A Piva. A DCT-domain system for robust image watermarking. Signal Processing, Vol. 66, 357-372, 1998

[9] Chen B and Wornell G W. Quantization index modulation: a class of provably good methods for digital watermarking and information embedding. IEEE Transactions on Information Theory, Vol. 47, No. 4, 1423-1443, 2001

[10] Bender, W Gruhl D, Morimoto N, Lu A. Techniques for data hiding. IBM Systems Journal, Vol. 35, No. 3, 313-336, 1996

[11] Nikolaidis N and Pitas I. Robust watermarking in the spatial domain. Signal Processing, Vol. 66, 385-403, 1998

[12] Yeo I K, Kim H J. Modified patchwork algorithm: a novel audio watermarking scheme, IEEE Transactions on Speech and Audio Processing, Vol. 11, No. 4, 381-386, 2003

[13] Coltuc D, Bolon P. Watermarking by histogram specification, Proceedings of SPIE, Vol. 3657, 252-263, Bellingham, Washington: SPIE Press, 1999

第 15 章　数据挖掘方法与技术

随着计算机技术和通信技术的快速发展，数据的获取、传输和处理构成了信息系统的核心环节。然而，商业企业、科研机构和政府部门普遍面临的一个问题是：数据的处理速度远远落后于数据的生成（获取）速度，这就导致在多年的信息化工作过程中积累了海量的数据资料，这些数据中隐藏着用以支撑我们判断形势并作出决策的依据，但由于高效数据处理机制的匮乏，当面对越来越多迅速膨胀的超大容量数据时，我们无法快速、准确地理解数据中包含的信息，难以获得有价值的知识。**数据挖掘**（Data Mining，DM）概念的提出，使我们有可能从海量的数据中发掘出隐藏其中的知识。

数据挖掘是目前数据库和信息决策领域的前沿研究方向之一，引起了学术界和工业界的广泛关注。IBM Almaden、GTE 等著名的工业研究实验室，以及 UC Berkeley 等众多的学术机构，都在这个领域开展了大量工作并取得了丰硕成果。本章主要介绍了数据挖掘的基本概念、基本原理和典型方法，并用实例说明了它在信息安全领域中的具体应用。

15.1　基本概念

人们将存储在数据库中的数据看作是形成知识的源泉，并将这种从数据中提取知识的过程形象地与矿石采掘工作联系起来，这也正是“数据挖掘”一词的由来。

定义 15.1.1　**数据挖掘**（data mining）是从大量的、不完全的、有噪声的、模糊的、随机的数据中，提取隐含在其中的、事先未知的、但又是潜在有用的信息和知识的过程。

数据挖掘是一门交叉学科，它汇聚了数据库、人工智能、统计学、可视化等不同学科和领域。自从 1995 年在加拿大蒙特利尔召开的第一届知识发现和数据挖掘国际学术会议以来，“数据挖掘”一词开始受到广泛的认可，随着研究工作的不断深入和扩展，其内涵也在不断丰富。

数据挖掘与数据库中的**知识发现**（knowledge discovery in database，KDD）存在着概念上的交叉，一种观点认为两者的概念是等同的，只是应用于不同的领域；另一种观点则认为数据挖掘是知识发现的一个核心环节。这里更倾向于后一种观点。按照学术界对于数据挖掘和知识发现这两个概念的描述来看，知识发现是把低级别的数据转化为高级别数据的抽象过程。知识可表示为概念（concepts）、规则（rules）、规律（regulations）、模式（patterns）等形式，而数据挖掘则是知识发现实现从数据到信息和知识转变的关键一步。因此，知识发现更多地体现出一种数据处理的目标，而数据挖掘则是为了达成这一目标而采取的某一环节的技术手段。

另一个需要澄清的概念是**联机分析处理**(onLine analytical processing,OLAP)。OLAP 主要通过多维的方式来对数据进行分析、查询和报表,主要用于对当前及历史数据进行分析、辅助领导决策。Gartner Group 等组织把 OLAP 视为数据挖掘的一部分,但更多的认识则把 OLAP 和数据挖掘当作互不相交的两部分。OLAP 可以帮助简化数据分析过程,其功能基本上是用户参与的汇总和比较;数据挖掘则可以自动发现隐藏在大量数据中的知识。另一方面,OLAP 大多是限于数据仓库中的数据;数据挖掘则既可以分析现存的、比数据仓库提供的汇总数据粒度更细的数据,也可以分析事务的、文本的、空间的和多媒体数据。

数据挖掘能够从海量数据中发现有效的、先前未知的并且最终可理解的信息。可以使用所抽取的信息来形成一个预测或分类模型,或者找出数据库记录间的相似性,发掘结果信息可帮助作出更准确的决策。

基本的知识发现过程如图 15.1 所示,它由 3 个核心环节组成。

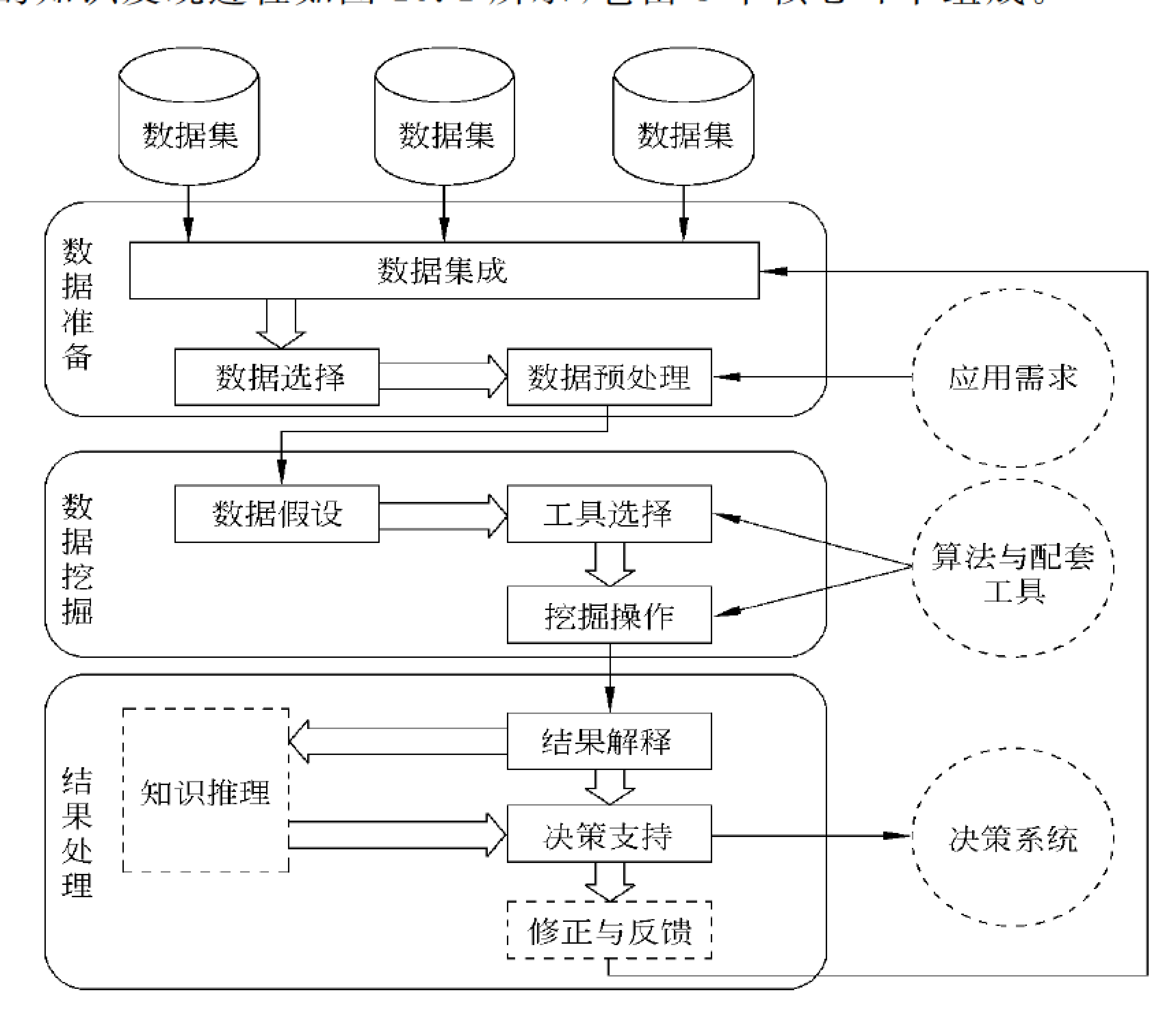


图 15.1　知识发现过程示意

(1) 数据准备:数据准备阶段包括数据集成、数据选择和数据预处理。其中,数据集成负责将多种来源的数据集进行合并处理,并对数据集进行修补以减少遗漏,实施数据清洗以减少可能影响挖掘结果的数据。数据选择确定需要实施挖掘分析的数据集合,提高数据挖掘的针对性。数据预处理则根据应用需求主要实现数据转换功能,以适应挖掘算法的要求,克服目前数据挖掘工具的一些局限性。

(2) 数据挖掘:数据挖掘阶段包括数据假设、工具选择和挖掘操作。其中,数据假设重点是决定如何产生假设,是让数据挖掘系统为用户产生假设,还是用户自己参

照数据库可能包含的知识提出假设。前一种称为发现型(discovery-driven)的数据挖掘,后一种称为验证型(verification-driven)的数据挖掘。工具选择是根据数据假设的结果,从算法库和工具库中确定实施本项知识发现过程所需的数据挖掘算法和配套工具,并由其完成核心的挖掘操作。算法和工具的选择将对后续挖掘结果的解释起到关键作用。

(3) 结果处理:结果处理阶段包括结果解释、知识推理、决策支持和修正与反馈等工作。重点是把提取的信息进行分析,通过决策系统提交给决策者。这一阶段不仅把结果表达出来,而且知识发现系统会采用解释和推理机制,将这些知识直接提供给决策者,也可以提供给领域专家,修正已有的知识库供系统共享。如果挖掘结果不够满意,需要重复以上知识发现的过程,实施有效的修正和反馈。

在上述知识发现的过程中,数据挖掘是其中的核心阶段。数据假设的确立和算法工具的选择将直接影响到整个知识发现过程是否能够从给定的数据中发掘出未知且有用的知识。下面将对数据挖掘的基本原理进行阐述。

15.2　基本原理

数据挖掘的初衷是帮助人们从海量的数据中提取出有用的、先前未知的知识,并指导作出合适的决策。数据挖掘与传统的统计学有着密切的联系。最初的一些数据挖掘方法直接来源于统计学。虽然统计学不能解决数据挖掘的所有问题,但却可以为数据挖掘提供基础的框架;同时,数据挖掘的出现也为统计学提供了一个崭新的应用领域,给统计学的理论研究提出了新的课题。

15.2.1　数据挖掘的任务

数据挖掘一般有以下 4 类主要任务。

定义 15.2.1　数据总结:数据总结是通过对数据库中数据的提炼,将数据从较低的表达层次抽象总结到较高的表达层次上,从而给出数据的综合描述,使分析人员实现对数据的总体把握。

常用的数据总结是利用统计学的方法,计算数据内容的均值、方差、总和、最大值、最小值等统计量,或者利用图形化的统计分析工具,形成描述数据特性的直方图、饼图、曲线图等。利用 OLAP 技术实现数据的多维查询也是一种常用的数据总结方法。

定义 15.2.2　数据分类(data classification):数据分类是通过提取数据库中数据项的特征属性,生成分类模型,该模型可以把数据库中的数据记录映射到给定类别中的一个。

分类模型主要用于实现对数据记录的类别预测,分类的应用非常广泛。例如,汽车生产商用于对购买不同型号汽车的客户群体进行分类,以区别出不同客户的喜好。另外,如医学诊断、行为判定、市场预测等,都广泛采用数据分类进行自动处理。

定义 15.2.3　关联分析(association analysis):关联分析是从已知的数据集中,产生数据项之间的关联规则,揭示不同数据项相互影响的程度。

关联分析的目的是揭示两个或多个变量的取值之间存在某种规律性。最简单的例子是：通过对顾客在超市购物记录的关联分析，可能会发现所有购买面包的顾客中有 90%的人同时购买了牛奶。这就说明面包和牛奶这两件商品之间从顾客的角度而言存在关联性，超市经营者可以考虑将其货柜调整到一起，以方便顾客的购物从而增加销售额。

序列模式分析同样也是试图找出数据之间的联系。但它的侧重点在于分析数据之间前后(因果)关系，因此对数据往往要求引入时间属性。序列模式分析非常适于寻找事物的发生趋势或重复性模式。

定义 15.2.4　聚类(clustering)：聚类是按照某种相近程度度量方法，将用户数据分成一系列有意义的子集合。每一个子集合中的数据性质相近，不同子集合之间的数据性质相差较大。

当要分析的数据缺乏描述信息，或者是无法组织成任何分类模式时，可以采用聚类分析。聚类和分类的区别是聚类不依赖于预先定义好的类，不需要训练集。

统计方法中的聚类分析是实现聚类的一种手段，它主要研究基于几何距离的聚类。人工智能中的聚类是基于概念描述的，即对某类对象的内涵进行描述，并概括这类对象的有关特征。

15.2.2　数据挖掘的方法

当前，国内外许多院校、科研机构、企业和学术团体都致力于数据挖掘技术的研究，开发了一系列实施数据挖掘的软件工具。主要的方法包括决策树、关联规则挖掘、神经网络、遗传算法及数据可视化、OLAP 等。

1. 决策树

决策树是用于实现数据分类的一种方法。通过针对已知训练数据集的学习，建立起用于对后续数据进行分类的决策树。在实施分类时则利用已建立的决策树，根据数据属性对数据所属的类型进行预测。决策树是采用直观的形式，将用于数据分类的规则进行了可视化表示，以提高数据分析过程的直观性和可理解性。决策树方法将在本章的第 3 节中详细介绍。

决策树方法也被广泛应用在针对安全数据的处理中，用以生成安全事件记录的分类模型并用于事件记录的分类。

2. 神经网络

神经网络建立在自学习的数学模型基础之上，它可以对大量复杂的数据进行分析，并可以完成对计算机来说极为复杂的模式抽取及趋势分析。

神经网络由一系列类似于人脑神经元一样的处理单元组成，称之为节点(node)。这些节点通过网络彼此互连，如果有数据输入，它们便可以进行确定数据模式的工作。神经网络由相互连接的输入层、中间层(或隐藏层)、输出层组成。中间层由多个节点组成，完成大部分网络工作。输出层输出数据分析的执行结果。例如，可以指定输入层为代表过去的销售情况、价格及季节等因素，输出层便可输出判断本季度的销

售情况的数据。

3. 关联规则挖掘

关联规则是一种简单却很实用的规则,它描述了一个事物中某些属性同时出现的规律和模式。例如,超级市场中通过 POS 系统收集存储了大量售货数据,记录了什么样的顾客在什么时间购买了什么商品,这些数据中常常隐含着关联规则。关联规则挖掘就是依据一定的置信度、支持度、期望置信度、作用度建立起关联规则的过程。典型的关联规则挖掘方法及上述概念的定义将在本章的第 3 节给出。

4. 遗传算法

遗传算法是一种基于生物进化论和分子遗传学的搜索优化算法。它首先将问题的可能解按某种形式进行编码,编码后的解称为染色体;随机选取 N 个染色体作为初始种群,再根据预定的评价函数对每个染色体计算适应值,性能较好的染色体有较高的适应值;选择适应值较高的染色体进行复制,并通过遗传算子,产生一群新的更适应环境的染色体,形成新的种群,直至最后收敛到一个最适应环境的个体,得到问题的最优解。

5. 联机分析处理

联机分析处理(OLAP)主要通过多维的方式来对数据进行分析、查询和报表。它不同于传统的联机事务处理(online transaction processing, OLTP)应用。OLTP 应用主要是用来完成用户的事务处理,如民航订票系统、银行储蓄系统等,通常需要进行大量的更新操作,同时对响应时间要求比较高。而 OLAP 应用主要是对用户当前及历史数据进行分析,辅助领导决策。其典型的应用有对银行信用卡风险的分析与预测、公司市场营销策略的制定等,主要是进行大量的查询操作,对时间的要求不太严格。

6. 数据可视化

对大批量数据的有效展现也是数据挖掘的重要方面。由于数据量很大,很容易使分析人员面对数据不知所措,数据挖掘的可视化工具可以为数据分析人员提供很好的帮助。

数据可视化工具大大扩展了传统商业图形的能力,支持多维数据的可视化,从而提供了多方向同时进行数据分析的图形方法。有些工具甚至提供动画能力,使用户可以观看到数据不同层次的细节。

15.3 典型的数据挖掘方法

15.3.1 关联分析

考虑一些涉及许多数据项的事务:事务 1 中出现了物品 A,事务 2 中出现了物品 B,事务 3 中则同时出现了物品 A 和 B。那么,物品 A 和 B 在事务中的出现相互之间是否有规律可循呢?在数据库的知识发现中,关联规则就是描述这种在一个事

务中物品之间同时出现的规律的知识模式。更确切地说，关联规则通过量化的数字描述物品 A 的出现对物品 B 的出现有多大的影响。

设 $R=\{I_1,I_2,\cdots,I_m\}$ 是一组数据项集，W 是一组事务集。W 中的每个事务 T 是一组数据项，且满足 $T\subseteq R$。假设有一个数据项集 X，一个事务 T，如果 $X\subseteq T$，则称事务 T 支持数据项集 X。

我们所要发掘的关联规则是指以下形式的一种数据隐含规则：

$X\Rightarrow Y$，其中 X，Y 是两组数据项，$X\subset T$，$Y\subset T$，$X\cap Y=\varnothing$

一般用以下 4 个参数来描述一个关联规则的属性。

定义 15.3.1　置信度(confidence)：置信度是指在出现了数据项集 X 的事务 T 中，数据项集 Y 也同时出现的概率，即 $P(Y|X)$。

如果在事务集 W 中所有支持数据项集 X 的事务中，有 $c\%$的事务同时也支持数据项集 Y，则 $c\%$称为关联规则 $X\Rightarrow Y$ 的置信度。如上面所举的牛奶和面包的例子，该关联规则的置信度就回答了这样一个问题：如果一个顾客购买了牛奶，那么他同时也购买面包的可能性有多大呢？在上述例子中，购买牛奶的顾客中有 90%的人购买了面包，所以置信度是 90%。

定义 15.3.2　支持度(support)：支持度描述了 X 和 Y 这两个数据项集的并集 C 在所有的事务中出现的概率，即 $P(X\cup Y)$。

如果 W 中有 $s\%$的事务同时支持数据项集 X 和 Y，则 $s\%$称为关联规则 $X\Rightarrow Y$ 的支持度。如果对现有的 1000 条销售记录进行统计，结果显示有 100 个顾客同时购买了牛奶和面包，那么上述关联规则的支持度就是 10%。

定义 15.3.3　期望置信度(expected confidence)：期望置信度描述了在没有任何条件影响时，数据项集 Y 在所有事务中出现的概率，即 $P(Y)$。

如果 W 中有 $e\%$的事务支持数据项集 Y，则 $e\%$称为关联规则 $X\Rightarrow Y$ 的期望置信度。如果对现有的 1000 条销售记录进行统计，结果显示有 200 个顾客购买了面包，则上述的关联规则的期望置信度就是 20%。

定义 15.3.4　作用度(lift)：作用度是置信度与期望置信度的比值，即 $P(Y|X)/P(Y)$。

作用度描述数据项集 X 的出现对数据项集 Y 的出现的影响程度。因为数据项集 Y 在所有事务中出现的概率是期望置信度；而数据项集 Y 在有数据项集 X 出现的事务中出现的概率是置信度，通过置信度对期望置信度的比值反映了在加入“数据项集 X 出现”的这个条件后，数据项集 Y 的出现概率发生了多大的变化。

置信度是对关联规则准确度的衡量，支持度则是对关联规则重要性的衡量。支持度说明了这条规则在所有事务中有多大的代表性，显然支持度越大，关联规则越重要。有些关联规则置信度虽然很高，但支持度却很低，说明该关联规则实用的概率很小，因此并不重要。

期望置信度描述了在没有数据项集 X 的作用下，数据项集 Y 本身的支持度；作用度描述了数据项集 X 对数据项集 Y 的影响力。作用度越大，说明数据项集 Y 受数据项集 X 的影响越大。一般情况下，有用的关联规则的作用度都应该大于 1，只有关

联规则的置信度大于期望置信度，才说明 X 的出现对 Y 的出现有促进作用，也说明了它们之间某种程度的相关性，如果作用度不大于 1，则此关联规则也就没有意义了。

关联分析的目的是从已知的事务集 W 中，产生数据项集之间的关联规则，保证其支持度和置信度大于用户预先指定的最小支持度（minimum support）和最小置信度（minimum confidence）。发掘关联规则通常可分为以下两个步骤进行。

第一步，从事务集 W 中找出所有支持度大于最小支持度的数据项集，称之为大数据项集（large itemsets），其他不满足支持度要求的数据项集则称为小数据项集（small itemsets）。这部分工作通常采用 Apriori、AprioriTid、AprioriHybrid 等算法来完成。

第二步，使用大数据项集产生期望的关联规则。产生关联规则的基本原则是其置信度必须大于预先指定的门限值，即最小置信度。

下面分别介绍这两个步骤所采用的具体算法。

1. 发掘大数据项集

这里首先给出一些重要概念的定义和表示方法。

定义 15.3.5　尺寸（size）：表示一个数据项集中包含的数据项数目。

k-itemset 表示尺寸为 k 的数据项集；$c[1]\cdot c[2]\cdot\cdots\cdot c[k]$分别表示包含 $c[1]$，$c[2]$，…，$c[k]$的 k-itemset，并且满足 $c[1]<c[2]<\cdots<c[k]$；L_k 表示第 k 轮计算过程得到的尺寸为 k 的大数据项集；C_k 表示第 k 轮计算过程得到的尺寸为 k 的候选数据项集（candidate itemsets）。

下面介绍两种典型的用于发掘大数据项的算法：Apriori 和 AprioriTid。

（1）Apriori 算法

图 15.2 给出了 Apriori 算法的流程。Apriori 算法根据轮循传递的原理来发现事务集中支持度大于最小支持度的大数据项集。每一轮中检查的数据项集所包含的数据项数目依次递增。

```
L1={large 1-itemsets};
for(k=2; Lk-1≠0; k++)do begin
    Ck=apriori-gen(Lk-1);
    forall transactions t∈W do begin
        Ct=subset(Ck,t);
        forall candidates c∈Ct do
            c.count++;
      end
      Lk={c∈Ck|c.count≥min sup};
end
Answer=∪K Lk
```

图 15.2　Apriori 算法流程

在第一轮，扫描整个事务集，对单个数据项的支持度进行计数，得到满足最小支持度要求的数据项。随后进行的每一轮（第 k 轮）计算过程分为两个步骤，首先根据上一轮（第 $k-1$ 轮）发现的大数据项集 L_{k-1} 为种子集合，调用 apriori-gen()函数产生

第 k 轮的候选数据项集 C_k；然后扫描整个事务集，对 C_k 中的每个候选集的实际支持度进行检查，得到本轮满足最小支持度要求的大数据项集 L_k，作为下一轮计算的种子集合，此步骤中需要调用 subset() 函数，用于得到 C_k 中所有被事务 t 包含的候选数据项集。如此反复进行轮循计算，直到没有新的大数据项集产生。此处不再给出 apriori-gen() 函数和 subset() 函数的详细流程。

(2) AprioriTid 算法

图 15.3 给出了 AprioriTid 算法的流程。该算法同样使用 apriori-gen 函数来根据上一轮得到的 L_{k-1} 产生本轮的候选数据项集 C_k。它的一个突出的优点是：在第一轮计算过程之后，将不再需要访问原始数据库来计算数据项集的支持度，而采用加上事务标识(TID)的候选数据项集 $\bar{C}_k$ 来代替。

```
1. L1={large 1-itemsets};
2. C̄k=database D;
3. for(k=2; Lk-1≠0; k++)do begin
4.    Ck= apriori-gen(Lk-1);
5.    C̄k=∅;
6.    forall entries t∈ C̄k-1 do begin
7.    Ct={c∈ Ck|(c-c(k))∈t.set-of-itemsets∧ (c-c(k-1))∈t.set-of-itemsets}
8.    forall candidates c∈ Ct do
9.        c.count++;
10.    if(Ct≠∅)then C̄k+=<t.TID,Ct>;
11.    end
12.    Lk={c∈ Ck|c.count≥min sup};
13. end
14. Answer=∪kLk
```

图 15.3　AprioriTid 算法流程

$\bar{C}_k$ 中的每个成员表示为〈TID:{X_k}〉的形式，每个 X_k 对应于一个在时间标识为 TID 的事务中存在的尺寸为 k 的大数据项集(large k-itemset)。当 $k=1$ 时，$\bar{C}_k$ 对应于整个原始数据库 D，仅仅将每个数据项 i 用长度为 1 的数据项集{i}来代替；当 $k>1$ 时，使用图中第 10 步的算法来生成 $\bar{C}_k$，$\bar{C}_k$ 中对应于事务 t 的成员是〈t.TID:{$c\in C_k|c\prec t$}〉，其中 $c\prec t$ 表示 c 被事务 t 包含。如果一次事务没有包含任何尺寸为 k 的候选数据项集，那么 $\bar{C}_k$ 中就没有对应于该事务的条目。

2. 产生规则

得到数据库中存在的大数据项集后，下一步需要生成满足条件的关联规则。生成关联规则的原则是：

对于每一个大数据项集 l，考虑 l 所有的非空子集 a。对每一个子集 a 来说，如果满足：

support (l)/support (a) >=minimum confidence(最小置信度)

那么就可以输出一条规则：$a \Rightarrow (l-a)$

可以采用深度优先的方式来递归地产生大数据项集的子集。例如，对于数据项集 ABCD，首先考虑子集 ABC，然后考虑子集 AB，…。如果大数据项集 l 的某个子集 a 不能产生满足置信度要求的规则，那么 a 的任意子集 b 也就不必考虑了。这是因为 support(b) >= support(a)，因此必然有：support(l)/support(b) <= support(l)/support(a)，即规则 $b \Rightarrow (l-b)$ 的置信度小于等于规则 $a \Rightarrow (l-a)$ 的置信度。算法流程见图 15.4。

```
forall large itemsets l_k, k≥2, do
  call genrules(l_k, l_k);
procedure genrules(l_k: large k-itemsets, a_m: large m-itemsets)
A={(m-1)-itemsets a_{m-1} | a_{m-1}⊂a_m};
forall a_{m-1}∈A do begin
  conf=support(l_k)/support (a_{m-1});
  if(conf≥minconf) then begin
    output the rule a_{m-1}⇒(l_k-a_{m-1}), with confidence=conf and support=support(l_k)
          if(m-1>1) then
              call genrules(l_k, a_{m-1});
    end
end
```

图 15.4　关联规则发掘算法流程 1

如果规则 $a \Rightarrow (l-a)$ 不成立，那么 a 的任意子集 b 也就不满足 $b \Rightarrow (l-b)$。因此，如果规则 $(l-c) \Rightarrow c$ 成立，那么 c 的任意子集 d 也就满足 $(l-d) \Rightarrow d$。基于这种考虑，如果已知尺寸为 $k-1$ 的数据项集的集合 H_{k-1}，规则 $(l-d) \Rightarrow d(\forall d \in H_{k-1})$ 满足置信度要求，那么可以使用 apriori-gen()函数来推出尺寸为 k 的候选数据项集的集合 H_k，再对 H_k 中的每个候选数据项集 e 检查其对应规则 $(l-e) \Rightarrow e(\forall e \in H_k)$ 的置信度，删除不符合置信度要求的数据项集，从而得到尺寸为 k 的满足要求的数据项集合 H_k。图 15.5 给出了该算法的流程。

15.3.2　序列挖掘

从本质上说，序列挖掘算法属于关联分析算法的范畴，其区别在于关联分析是发掘数据记录中不同数据项之间的横向关联性，而序列挖掘则是发现不同数据记录之间的纵向相关性。序列挖掘的目标是在事务数据库中发掘出序列模式(large sequences)，即满足用户指定的最小支持度要求的大序列，并且该序列模式必须是最高序列(maximal sequence)。

这里首先介绍序列挖掘中的一些重要定义。

定义 15.3.6　序列(sequence)：序列表示按顺序排列的一组数据项集。

```
1. forall large k-itemsets l_k, k≥2 do begin
2.   H_1 = {consequents of rules derived from l_k with one item in the consequent};
3.   call ap-genrules(l_k, H_1);
4. end

procedure ap-genrules(l_k: large k-itemsets, H_m: set of m-item consequents
    if(k>m+1)then begin
        H_{m+1} = apriori-gen(H_m);
        forall h_{m+1} ∈ H_{m+1} do begin
            conf= support(l_k)/support (l_k - h_{m+1});
            if(conf≥minconf)then
                output the rule (l_k - h_{m+1}) ⇒ h_{m+1} with confidence= conf and
support= support(l_k);
            else
                delete h_{m+1} from H_{m+1};
        end
        call ap-genrules(l_k, H_{m+1});
end
```

图 15.5 关联规则发掘算法流程 2

定义 15.3.7 序列的包含(contain)：对序列 A:$\{a_1a_2\cdots a_n\}$和序列 B:$\{b_1b_2\cdots b_m\}$，如果存在整数 $i_1<i_2<\cdots<i_n$，使得 $a_1\subseteq b_{i_1}$，$a_2\subseteq b_{i_2}$，$\cdots$，$a_n\subseteq b_{i_n}$，就称序列 A 被序列 B 包含，记为 $A\prec B$。

定义 15.3.8 最高序列(maximal sequence)：没有被其他任何序列所包含的序列。

定义 15.3.9 数据项集的支持度(support for an itemset)：在一次事务中包含了该数据项集中所有数据项的主体在全部主体中所占的比例。

定义 15.3.10 大数据项集(litemset，large itemset)：满足最小支持度要求的数据项集。

定义 15.3.11 大序列(large sequence)：一组大数据项集的列表，大序列中的每一个数据项集都必须是大数据项集。

挖掘序列模式通常分为以下 5 个步骤进行。

(1) 排序阶段：以事务的主体为主键，事务时间为次键，对原始数据库进行排序，将其转换为主体序列的数据库。

(2) 大数据项阶段：找出所有的大数据项集 L(此过程也相当于找出了所有长度为 1 的大序列)，并把大数据项集映射为一组相邻的整数，每个大数据项集对应一个整数。

(3) 转换阶段：将数据库中主体序列的每一次事务用该事务包含的大数据项集(映射的整数)代替。

(4) 序列阶段：利用大数据项集发掘序列模式，该阶段是序列挖掘的关键步骤。

(5) 序列最高化阶段：找出所有序列模式的最高序列集。

按照计数方式的不同，序列挖掘算法分为以下两个大类：一类称为 Count-all 算法，即通过对所有的大序列(包括非最高序列)进行计数来计算支持度，代表算法是 AprioriAll；另一类称为 Count-some 算法，即通过避免或减少对那些被更长序列所包含的序列进行计数来提高系统性能，代表算法是 AprioriSome 和 DynamicSome。下面简单介绍这几种典型的序列挖掘算法。

1. AprioriAll 算法

算法流程如图 15.6 所示。在每一轮计算过程中，使用上一轮得到的大序列产生本轮的候选大序列(candidate sequences)，再访问事务数据库，计算候选大序列的支持度，得到本轮的计算结果。

```
L1={large 1-sequences};
for  (k=2; Lk-1≠∅;k++)do
       begin
       Ck=apriori-candidate-generation(Lk-1) ;
       foreach customer-sequence c in the database do
              Increment the count of all candidates in Ck that are contained in c
       Lk=Candidates in Ck with minimum support
       end
Answer=Maximal sequences in ∪k Lk
```

图 15.6 AprioriAll 算法流程

从图中可以看出，该算法中需要完成两个重要函数：Apriori-candidate-generation()函数输入上一轮得到的大序列 L_{k-1}，输出本轮的候选大序列 C_k；Subsequence()函数输入候选大序列集 C_k 和客户序列 c，输出 c 中包含的 C_k 中的候选大序列，用于计算候选大序列的支持度。

2. AprioriSome 算法

AprioriAll 算法在每一轮中生成候选大序列集 C_k 后，都要访问数据库对 C_k 中的序列进行支持度检查，如果数据库记录很多的话，这将花费大量的时间。基于这种考虑，人们提出了一种 AprioriAll 的改进算法——AprioriSome。

AprioriSome 算法将计算过程分为两个阶段：前向阶段(forward phase)用于找出指定长度的所有大序列；后向阶段(backward phase)则用于查找其他长度的所有大序列。图 15.7 给出了 AprioriSome 算法的流程。

3. DynamicSome 算法

DynamicSome 算法与 AprioriSome 算法类似，在前向阶段跳过了对某些长度的候选序列进行访问数据库、计算支持度的工作。对哪些长度的候选序列进行计数取决于预先指定的步长(step)。DynamicSome 算法流程如图 15.8 所示。

```
// Forward Phase
L_1 = {large 1-sequences};
C_1 = L_1;
Last=1;
for (k=2; C_{k-1}≠∅ and L_last≠∅; k++) do
  begin
  if (L_{k-1} known) then
      C_k=apriori-candidate-generation (L_{k-1});
  else
      C_k=apriori-candidate-generation (C_{k-1});
  if (k==next(last)) then begin
    foreach customer-sequence c in the database do
            Increment the count of all candidates in C_k that are contained in c
            L_k=Candidates in C_k with minimum support
            last=k
      end
  end

//Backward Phase
for (k--; k>=1; k--) do
if (L_k was not determined in the forwared phase) then begin
    delete all sequences in C_k contained in some L_i, i>k;
     foreach customer-sequence c in D_T do
            Increment the count of all candidates in C_k that are contained in c
    L_k=Candidates in C_k with minimum support
    end
else begin
    delete all sequences in L_k contained in some L_i, i>k
    end
Answer=∪_k L_k
```

图 15.7 AprioriSome 算法流程

```
X_k=subseq(L_k, c);
forall sequences x∈X_k do
    x.end=min{j|x≺{c_1 c_2 … c_j}};
X_j=subseq(L_j, c);
forall sequences x∈X_j do
    x.start=max{j|x≺{c_j c_{j+1} … c_n}};
Answer=join of X_k with X_j with the join condition X_k.end < X_j.start
```

图 15.8 DynamicSome 算法流程

15.3.3　数据分类

数据分类的目的是提取数据库中数据项的特征属性，生成分类模型，该模型可以把数据库中的数据记录映射到给定类别中的一个。典型的数据分类处理步骤如下：

（1）获得训练数据集（training set），该数据集中的数据记录具有和目标数据库中数据记录相同的数据项。

（2）训练数据集中每一条数据记录都有已知的类型标识与之相关联。

（3）分析训练数据集，提取数据记录的特征属性，为每一种类型生成精确的描述模型。

（4）使用得到的类型描述模型对目标数据库中的数据记录进行分类或生成优化的分类模型（分类规则）。

构造数据分类模型的具体方法有很多，如统计学、机器学习、神经网络、专家系统等。下面对其中典型的基于决策树及基于统计的分类方法进行介绍。

1. 基于决策树的分类

决策树是用于实现数据分类的常用方法。在决策树的每个内部节点上选用一个属性进行分割，每个分支代表一个测试输出，决策树的叶子节点表示一个类分布。例如，给出如表 15.1 所示的示例数据记录。

表 15.1　数据分类示例数据记录

年　龄	薪　水	类　型	年　龄	薪　水	类　型
30	65	G	55	40	B
23	15	B	55	100	G
40	75	G	45	60	G

根据上述记录可以生成用于分辨记录类型的决策树如图 15.9 所示。

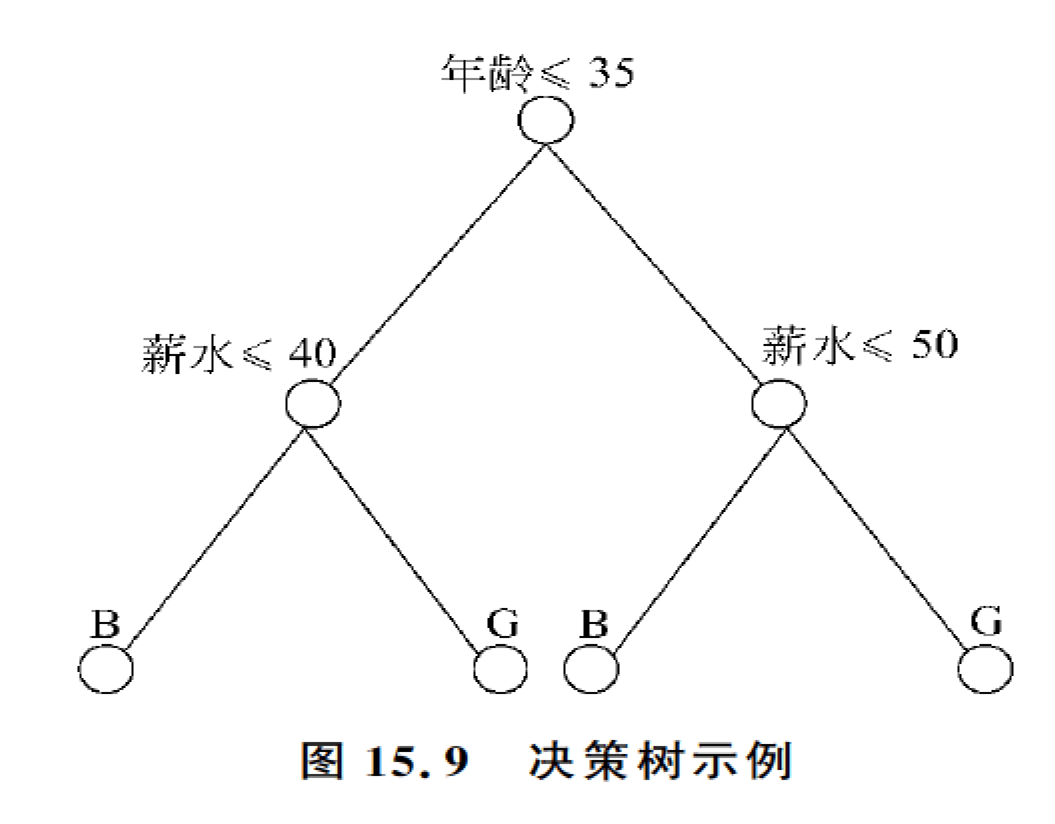


图 15.9　决策树示例

使用决策树方法进行数据分类时，首先利用训练样本，在开始阶段将数据设置在根节点，采用递归方式进行数据分割，基于启发式规则或统计度量来选择分类属性，生成决策树模型。在此基础上去除一些可能是噪声或者异常的数据，从而实现决策树剪枝。针对未知类型的数据记录，按照决策树采用的分割属性逐层往下，直至到达某个叶子节点，从而使用决策树对该数据记录进行分类。

数据分割过程中停止分割的条件是：一个节点上的数据都属于同一个类别，并且没有属性可以再用于对数据进行分割。在数据分类算法中，属性的选择是至关重要的，它关系到是否能够生成准确描述分类特

征的决策树。属性选择度量标准主要有以下 3 类。

(1) 信息增益(information gain)

以 ID3 算法为代表,其基本思想是:决策树的每个节点对应一个非类别属性,每条边对应该属性的每个可能值。以信息熵的下降速度作为选取测试属性的标准,即所选的测试属性是从根到当前节点的路径上尚未被考虑的具有最高信息增益的属性。

(2) 增益比率(gain ration)

以 C4.5 算法为代表,其基本思想是:利用增益比率的概念,合并连续值属性。该方法可以处理缺少属性值的训练样本,并通过使用不同的剪枝技术以避免树的不平衡。

(3) 基尼指数(gini index)

以 SLIQ、SPRINT 算法为代表,其基本思想是:假设数据集 T 包含来自 N 个类的样本,通过预定义的基尼指数,计算某个划分将数据集 T 分成两个子集 S_1 和 S_2 后的分割基尼指数,提供最小分割基尼指数的就被选择作为分割的标准。

在基于决策树的分类方法中,树的剪枝问题是需要重点关注的。剪枝的目的是消除决策树的过适应问题。

定义 15.3.12 过适应(overfitting):过适应是指推出过多的假设与训练数据集相一致,导致所作出的假设泛化能力过差。

剪枝的实质是消除训练集中的异常和噪声。实现树剪枝的方法主要有先剪枝法(即提前停止树的构造)和后剪枝法(即树完全生长后再剪枝)两种。剪枝标准可以采用最小描述长度原则(MDL),因为根据 MDL 原则,对数据进行编码的最佳模型是使得用该模型描述数据和描述这个模型的代价的和最小的模型,因此在剪枝时对决策树进行二进制编码,编码所需二进制位最少的树即为“最佳剪枝树”;剪枝标准也可以采用期望错误率最小原则,即对树中的内部节点计算其剪枝/不剪枝可能出现的期望错误率,比较后加以取舍。

2. 基于统计的分类

基于决策树的分类是一种确定性分类问题,即可以唯一地确定一条数据记录所属的类型,但在实际情况中,往往会出现类别重叠现象,此时的数据分类问题就转变为预测给定样本属于一个特定类的概率。朴素贝叶斯分类算法就是解决这一问题的典型方法。

朴素贝叶斯分类是假定一个属性值对给定类的影响独立于其他属性的值,预测未知样本的类别为后验概率最大的那个类别。贝叶斯定理提供了后验概率的计算方法,即

$$P(H \mid X) = \frac{P(X \mid H)P(H)}{P(X)}$$

其中 $P(X)$ 为 X 的先验概率;$P(H)$ 为 H 的先验概率;$P(X|H)$ 为条件 H 下 X 的后验概率;$P(H|X)$ 为条件 X 下 H 的后验概率。

设样本有 n 个属性 $(A_1, A_2, \cdots, A_n)$,每个样本可看作是 n 维空间的一个点

$X=(x_1,x_2,\cdots,x_n)$。假定有 m 个不同的类别 $C_1,C_2,\cdots,C_m$。X 是一个未知类别的样本。

预测 X 的类别为后验概率最大的那个类别，即算法将未知类别的样本 X 归到类 C_i，当且仅当 $P(C_i|X)>P(C_j|X)$ 对于所有的 j 成立$(1\leqslant j\leqslant m,j\neq i)$，即 $P(C_i|X)$ 最大。

根据贝叶斯定理得知：$P(C_i|X)=P(X|C_i)P(C_i)/P(X)$。由于 $P(X)$对于所有类为常数，因此只需 $P(X|C_i)P(C_i)$取最大值即可。类的先验概率 $P(C_i)$由 $P(C_i)=S_i/s$ 估算，其中 S_i 为训练样本中属于类 C_i 的样本数，s 为全部训练样本的样本数；$P(X|C_i)$则由 $P(X|C_i)=P(x_1|C_i)P(x_2|C_i)\cdots P(x_n|C_i)$估算。

对未知样本 X 分类，对每个类 C_i 都计算 $P(X|C_i)P(C_i)$。样本 X 被指派到类 C_i，当且仅当 $P(X|C_i)P(C_i)>P(X|C_j)P(C_j)$，$(1\leqslant j\leqslant m,j\neq i)$即 X 被指派到其 $P(X|C_i)P(C_i)$最大的类 C_i。

15.3.4　聚类

聚类也称为簇(cluster)，是指一个数据对象的集合。其特点是在同一个类中的对象之间具有相似性，而在不同类的对象之间是相异的。聚类分析就是把一个给定的数据对象集合分成不同的簇的过程。

聚类是一种无监督分类法，没有预先指定的类别。其典型的应用是作为一个独立的分析工具，用于了解数据的分布，或作为其他算法的一个数据预处理步骤。聚类分析在市场销售、土地使用、保险、城市规划、地震研究等各个领域获得了广泛的应用。

一个好的聚类方法要能产生高质量的聚类结果——簇。这些簇要具备两个特点，即高的簇内相似性和低的簇间相似性。聚类结果的好坏取决于该聚类方法采用的相似性评估方法以及该方法的具体实现，聚类结果的好坏还取决于该聚类方法是能发现某些还是所有的隐含模式。通常来说，聚类算法需要满足以下特性：

- 可伸缩性；
- 能够处理不同类型的属性；
- 能发现任意形状的簇；
- 在决定输入参数时，尽量不需要特定的领域知识；
- 能够处理噪声和异常；
- 对输入数据对象的顺序不敏感；
- 能处理高维数据；
- 能产生一个好的、能满足用户指定约束的聚类结果；
- 结果是可解释的、可理解的和可用的。

对于聚类算法而言，如何衡量两个对象之间的相似度(相异度)是至关重要的。通常使用距离来进行衡量。对不同类型的变量，距离函数的定义通常是不同的，而且，根据实际的应用和数据的语义，在计算距离时，不同的变量有不同的权值相联系。常用的距离度量方法如下。

(1) 明考斯基距离

$$d(i,j)=\sqrt[q]{(|x_{i_1}-x_{j_1}|^q+|x_{i_2}-x_{j_2}|^q+\cdots+|x_{i_p}-x_{j_p}|^q)}$$

其中,$i=(x_{i_1},x_{i_2},\cdots,x_{i_p})$ 和 $j=(x_{j_1},x_{j_2},\cdots,x_{j_p})$是两个 p 维的数据对象,q 是一个正整数。

(2) 曼哈顿距离

当上述公式中的 $q=1$ 时,此时得到的计算结果称为曼哈顿距离,即

$$d(i,j)=|x_{i_1}-x_{j_1}|+|x_{i_2}-x_{j_2}|+\cdots+|x_{i_p}-x_{j_p}|$$

(3) 欧几里得距离

当上述公式中的 $q=2$ 时,此时得到的计算结果称为欧几里得距离,即

$$d(i,j)=\sqrt{(|x_{i_1}-x_{j_1}|^2+|x_{i_2}-x_{j_2}|^2+\cdots+|x_{i_p}-x_{j_p}|^2)}$$

距离函数有以下特性:

- $d(i,j)\geqslant 0$
- $d(i,i)=0$
- $d(i,j)=d(j,i)$
- $d(i,j)\leqslant d(i,k)+d(k,j)$

可以根据每个变量的重要性赋予一个权重来帮助提高聚类的准确性。

15.4 应用举例

数据挖掘目前在医学、经济学、地质学、气象学等多个领域均有成功的应用案例。随着越来越多的业务需求被不断明确,数据挖掘应用的领域和解决的问题会越来越广泛。一些应用系统,如 ERP、SCM、HR 等系统也逐渐与数据挖掘集成起来,用以提高系统的决策支持能力。这方面的研究热点包括数据挖掘与商业智能(BI)、CRM、WEB 应用的结合。

数据挖掘技术在信息安全领域也得到了广泛的应用,由于其具有对海量数据进行智能化分析并从中挖掘出有价值信息的特性,而海量数据的处理正是众多安全审计系统和入侵检测系统所普遍面临的难题,因此在这两项安全技术上率先得到了应用。本节将简单介绍数据挖掘技术在这两方面的一些应用现状。

入侵检测(intrusion detection)技术实质上归结为对安全审计数据的处理。这种处理可以针对网络数据,也可以针对主机的审计记录、应用程序的日志文件或其他类型的审计数据。安全事件审计系统作为保障信息系统安全的基础部件,已越来越多地被应用到操作系统和网络安全管理工具中。针对安全事件审计数据的分析处理,其目的正是为了发现系统或用户行为的异常。然而,操作系统的日益复杂化和网络数据流量的急剧膨胀,导致了安全审计数据同样以惊人的速度激增。激增的数据背后隐藏着许多重要的信息,人们希望能够对其进行更高抽象层次的分析,以便更好地利用这些数据。目前的审计系统可以高效地实现安全审计数据的输入、查询、统计等功能,但无法发现数据中存在的关联、关系和规则,无法根据现有的数据预测未来的

发展趋势，缺乏挖掘数据背后隐藏的知识的手段，导致了“**数据爆炸但知识贫乏**”的现象。

如何从大量的审计数据中提取出具有代表性的系统特征模式，用于对程序或用户行为作出描述，是实现安全事件审计系统的关键。为了对审计数据进行全面、高速和准确的分析，需要利用如数据挖掘、机器学习等智能方法来处理安全事件数据，从包含大量冗余信息的数据中提取出尽可能多的隐藏的安全信息，抽象出利于进行判断和比较的特征模型，这种特征模型可以是基于误用检测（misuse detection）的特征向量模型，也可以是基于异常检测（anomaly detection）的行为描述模型。根据这些特征向量模型和行为描述模型，可以由计算机利用相应的算法判断出当前网络或系统行为的性质。

数据挖掘本身是一项通用的知识发现技术，其目的是要从海量数据中提取出感兴趣的数据信息（知识）。**将数据挖掘技术应用于入侵检测领域，利用数据挖掘中的关联分析、序列挖掘等算法提取与安全相关的系统特征属性，根据系统特征属性生成安全事件的分类模型或进行安全事件聚类分析，从而实现针对安全事件的自动鉴别**。基于数据挖掘的入侵检测方法包括了针对安全事件审计数据的数据采集、数据准备和预处理、特征变量选取、数据挖掘、挖掘结果处理以及结果可视化等一系列的过程，其中数据挖掘是整个过程的关键。

目前对挖掘算法的研究已经比较成熟，有许多算法可以使用。然而，真正要从海量数据中提取出感兴趣的数据信息（知识），需要强调的一点就是“**特定应用**”。算法实现必须建立在特定应用的基础之上，并且需要具有足够的先验知识。实验表明，对系统安全的先验知识往往体现在对原始数据中有价值的变量集的选择上，而这往往是系统实现中最大的难点。

从入侵检测的角度来看，数据挖掘可以看作是入侵检测的手段，但是当入侵检测的焦点归结为安全审计数据的分析问题时，数据挖掘就成为一种目的，关注的焦点集中在从大量的、包含冗余信息的数据中发掘出隐藏的、先前未知的知识。此时，就不再局限于传统数据挖掘所定义的关联、分类、聚类等算法，而可以将来自于其他人工智能、机器学习领域的算法统一地包括进来。因此，下面将介绍几种在入侵检测领域获得应用的方法，包括数据挖掘、神经网络、人工免疫等。

15.4.1　基于数据挖掘的入侵检测

美国哥伦比亚大学的 Wenke Lee 在其完成的博士论文[4]中，提出了将数据挖掘技术应用到入侵检测中，详细阐述了针对基于网络的审计数据和基于主机的系统调用数据，利用数据挖掘算法进行特征提取、相关分析和智能化的数据分类等方法，以此产生用于入侵检测的误用检测规则或异常检测模型。Wenke Lee 利用数据挖掘中的关联算法和序列挖掘算法提取用户的行为模式，利用分类算法对用户行为和特权程序的系统调用进行分类预测。实验结果表明，这种方法在入侵检测领域有很好的应用前景。

Wenke Lee 所在的研究组在 1998 年参加了由美国国防部高级研究计划署

(DARPA)资助的入侵检测评估(intrusion detection evaluation)计划。在这次评估计划中,由 MIT 的 Lincoln 实验室提供了在模拟军事网络环境中记录的 7 个星期的网络流量和主机系统调用记录日志,这些数据全部采用 tcpdump 和 Solaris BSM 审计数据的格式提供,其中包括了大约 500 万次会话,其中包含上百种攻击。这些攻击可以分为 4 种主要类型。

- 拒绝服务攻击:如 Ping of Death、teardrop、smurf、SYN Flood 等。
- 远程攻击(R2L):如基于字典的口令猜测和缓冲区溢出攻击。
- 本地用户非法提升权限的攻击(U2R)。
- 网络扫描:包括端口扫描和漏洞扫描。

Wenke Lee 研究组分别从网络和主机两方面进行了审计数据的挖掘处理。

1. 针对网络数据

其主要做法是使用网络服务端口(service)作为网络连接记录的类型标识,根据大量的正常连接记录生成各个服务类型的分类模型,在测试过程中根据分类模型对当前的连接记录进行分类,并与实际服务类型进行比较,从而判断出该分类模型的准确性。

2. 针对主机数据

使用了一种快速的规则学习算法 RIPPER,通过对正常调用序列的学习来预测随后发生的系统调用序列,并对结果进行了进一步的抽象分析以降低算法的预测误差。

根据 DARPA 的报告,由哥伦比亚大学实现的基于数据挖掘的入侵检测系统在检测拒绝服务攻击和扫描方面优于其他系统,在检测本地用户非法提升权限方面与其他系统大概持平,在检测远程攻击如缓冲区溢出方面,所有的系统表现都不令人满意,检全率都在 70%以下。图 15.10 显示的是 4 种参加测试的系统对拒绝服务攻击的检测结果,横坐标是误报率,纵坐标是检全率。

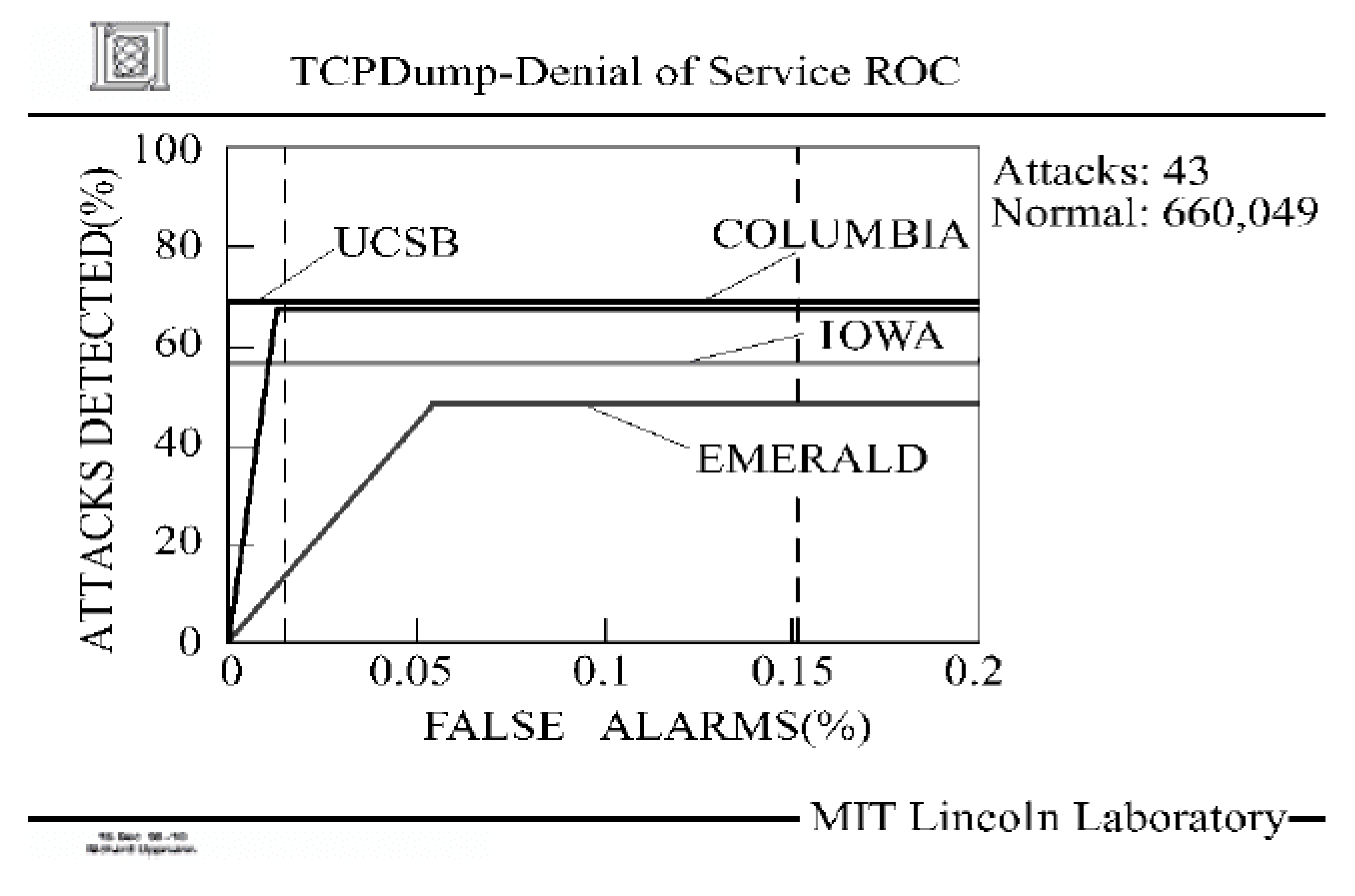


图 15.10　1998 年入侵检测评估结果

Wenke Lee 另一个突出的贡献是提出并验证了将信息论中“熵”(entropy)的概念引入安全领域,以解决入侵检测系统中特性属性的选择问题,用于构建检测模型,

这项工作将在本节的后续部分进行介绍。Wenke Lee 还对基于数据挖掘的入侵检测系统在配置到实时环境中时出现的问题进行了深入的研究，包括检测的准确性、检测效率和系统的实用性[5]。

哥伦比亚大学数据挖掘实验室的 Leonid Portnoy 则使用了数据挖掘中的聚类算法，通过计算和比较记录间的矢量距离，对网络连接记录、用户登录记录进行自动聚类，从而完成对审计记录是否正常的判断工作[6]。Leonid Portnoy 使用的数据是 KDDCUP99。

15.4.2 基于神经网络的入侵检测

作为人工智能的一个重要分支，神经网络(neural network)在入侵检测领域得到了很好的应用，它使用自适应学习技术来提取异常行为的特征，需要对训练数据集进行学习以得出正常的行为模式。这种方法要求保证用于学习正常模式的训练数据的纯洁性，即不包含任何入侵或异常的用户行为。

神经网络由大量的处理元件组成，这些处理元件称之为"单元"(units)，单元之间通过带有权值的"连接"(connections)进行交互。网络所包含的知识体现在网络的结构(单元之间的连接、连接的权值)中，学习过程也就表现为权值的改变和连接的添加或删除。

神经网络的处理包含两个阶段。第一阶段的目的是构造入侵分析模型的检测器，使用代表用户行为的历史数据进行训练，完成网络的构建和组装；第二阶段则是入侵分析模型的实际运作阶段，网络接收输入的事件数据，与参考的历史行为相比较，判断出两者的相似度或偏离度。在神经网络中，使用以下方法来标识异常的事件：改变单元的状态、改变连接的权值、添加连接或删除连接，同时也提供对所定义的正常模式进行逐步修正的功能。

神经网络方法对异常检测来说，具有很多优势：由于不使用固定的系统属性集来定义用户行为，因此属性的选择是无关的；神经网络对所选择的系统度量(metrics)也不要求满足某种统计分布条件，因此与传统的统计分析相比，具备了非参量化统计分析的优点。

另一方面，将神经网络应用在入侵检测中，也存在一些问题。例如，在很多情况下，系统趋向于形成某种不稳定的网络结构，不能从训练数据中学习到特定的知识，这种情况目前尚不能完全确定产生的原因。其次，神经网络对判断为异常的事件不会提供任何解释或说明信息，这导致了用户无法确认入侵的责任人，也无法判定究竟是系统哪方面存在的问题导致了攻击者得以成功入侵。另外，将神经网络应用于入侵检测，其检测的效率问题也是需要解决的。

Tulane 大学的 David Endler 针对 Solaris 系统的 BSM 模块所产生的系统调用审计数据使用神经网络进行机器学习。Anup K. Ghosh 也采用针对特定程序的异常检测，建立软件程序的进程级行为模式，通过区分正常软件行为和恶意软件行为来发现异常。使用预先分类的输入资料对神经网络进行训练，学习出正常和非正常的程序行为[7]。

15.4.3 基于人工免疫的入侵检测

美国新墨西哥大学的 Stephanie Forrest 提出了将生物免疫机制引入计算机系统的安全保护框架中。免疫系统最基本也是最重要的能力是识别“自我/非自我”(self/nonself),它能够识别哪些组织是属于正常机体的,不属于正常的就认为是异常,这个概念和入侵检测中异常检测的概念非常相似。

免疫系统能够保护生命机体不受病原体的侵害,这种作用与安全系统在计算机系统中的作用类似。研究人员注意到:免疫系统对病原体的检测是相当精确的,这与计算机安全系统的不可靠形成了鲜明的对比。免疫系统具有以下一些重要的特征。

1. *分层保护*

大自然赋予生命多层保护机制,如皮肤、生理环境(pH 和体温)、受激反应等,而计算机安全系统往往是单层次的。

2. *分布式检测*

免疫系统的检测是高度分布的,没有一个统一的中心控制,这种分布机制保障了系统的高度可靠性。

3. *各组成部分的相互独立性*

这种机制的优点是即使某一个方面的保护失效,也不会影响系统的其他部分。

4. *能够检测未知*

免疫系统不但能够记忆曾经感染过的病毒的特征,还能够有效地检测未知的病毒,这种能力是绝大多数计算机安全系统所缺乏的。

从免疫学的观点来看,免疫系统最基本也是最重要的能力不是识别异常,而是能够识别自我。其关键点是:**使用一组稳定的特征来定义自我**。很显然,对于计算机安全系统而言,要解决这个问题相当困难,第一,恶意代码隐藏在正常代码之中难以区分,第二,系统可能的状态几乎是无限的,寻找一组稳定的特征来定义自我并不容易。

新墨西哥大学的研究小组通过大量的实验发现:对一个特定的程序来说,其系统调用序列是相当稳定的。使用系统调用序列来识别“自我”,应该可以满足系统的要求。在这个假设的前提下,该研究小组提出了基于系统调用的短序列匹配算法,并作了大量开创性的工作[8]。

15.4.4 应用于入侵检测的数据源分析

前面曾经强调过,真正要从海量数据中提取出感兴趣的数据信息(知识),需要强调的一点就是“特定应用”。算法实现必须建立在特定应用的基础之上,并且需要具有足够的先验知识。另一方面,如何选择数据源进行挖掘,保证数据源能够提供特定应用所需的各项特征,并充分发挥算法的效果,从而能够得到用于对数据记录进行准确分类、关联和聚类等操作的模型,同样是将数据挖掘应用于入侵检测时所面临的基础性问题。本小节将说明可应用于数据挖掘的各类入侵检测数据源,并介绍一种可

用于判断数据源质量的方法。

1. 网络数据源

就网络数据而言，可以分为多个分析层次。例如，land① 攻击和 teardrop② 攻击通过对 IP 报头的分析就可以准确检测，而端口扫描就必须通过维护一个基于时间窗口的状态栈，通过对多数据包的报头进行分析才能检测。对于其他一些涉及高层协议的攻击方法，比如 mail bomb、cgi 漏洞攻击，则必须进行相应的高层协议解析才能检测。正因为如此，一个好的网络入侵检测系统总是尽可能多地解析高层协议，或者根据已知攻击模式的特征进行尽可能的准确匹配，或者通过建立正常的协议行为模式以检测异常。

网络数据源通常具有适用于检测基于网络的攻击行为、应用范围广、扩展性好等优点，但同时由于其本身的缺陷，也存在着诸如资源消耗大、检测复杂攻击的准确率低、无法处理加密数据等缺陷。

2. 主机数据源

安全审计数据首先来自于主机日志，在 UNIX 环境下，通常指由后台进程 syslogd 产生的 syslog 日志，以及由其他应用程序如 Apache Web Server 或 Ftp 产生的日志，如 access_log、xferlog、maillog 等。毫无疑问，从这些日志中可以得到大量有价值的信息。事实上，经常检测日志是维护系统安全不可缺少的一个环节。但是，这些日志基本上都是以方便人阅读的格式存放的，如何将数据挖掘算法应用于日志数据还存在很多的困难。

国外的研究者发现，特权进程的系统调用序列能够较好地满足审计数据源的要求。在 UNIX 环境下，所谓特权进程，是指能够以 root 权限执行的进程，如 Sendmail 和 Syslogd。特权进程通常是攻击的重点目标。系统调用发生在用户进程（比如 emacs）通过特殊函数（如 open）请求内核提供服务时，在系统调用层次上，可以利用数据挖掘方法生成分别用于误用检测和异常检测的模型。

主机数据源通常具有检测的针对性强、准确率高、与操作系统结合紧密、适于检测复杂攻击模式等优点，但由于依赖于本机的操作系统，因此也存在着诸如只能检测针对本机的攻击、不适合检测基于网络协议的攻击等局限性。

可以说，网络数据源和主机数据源各有优势，又各有不足。值得注意的是，两者在各自所擅长的检测领域上存在互补性。因此，最佳的安全审计数据处理方案应该是综合这两方面的审计数据源。下面介绍一种用于对数据源质量进行分析的方法。

3. 数据源质量分析

Wenke Lee 提出了利用信息论的某些概念：熵、条件熵、相对熵和信息增益，可以定量地描述一个数据集的特征，分析数据源的质量，从而为模型的选择提供理论依据[9]。以下给出一些用于数据源质量分析的定义及其具体的应用。

① land：攻击者通过发送源 IP 地址和目的 IP 地址相同的数据包来造成对目标主机的拒绝服务。

② teardrop：攻击者通过发送无法重组的错误 IP 分片造成目标主机的拒绝服务。

给定数据集合 X，其熵 $H(X)=\sum_{x\in C_X}P(x)\log\frac{1}{P(x)}$。一般地说，熵的值越小说明数据的分布越均匀。例如，如果所有的数据项都属于同一个类，那么熵等于 0；熵越大说明数据分布越不均匀。对于异常检测来说，熵可以作为衡量数据规则程度的一个度量。在安全审计数据中，每一个唯一的记录代表一个类，熵越小表示不同的记录数目越少，数据也就越规则，从而通过已知预测未知的可靠性越大，通过这样的数据集合建立的检测模型的可信度就越高。

安全审计数据通常都具有时间上的序列特征，条件熵可以用来衡量这种特征。令 $X=(e_1,e_2,\cdots,e_n)$，令 $Y=(e_1,e_2,\cdots,e_k)$，其中 $k<n$；条件熵 $H(X\mid Y)=\sum_{x,y\in C_x,C_y}P(x,y)\log\frac{1}{P(x\mid y)}$ 可以衡量在给定 Y 以后，剩下的 X 的不确定性还有多少。条件熵越小表示不确定性越小，根据这样的数据集合建立的检测模型的准确性越好。

在异常检测中，可以使用数据挖掘方法，利用一组训练数据来构造模型，再将这个模型运用到测试数据中。如果希望模型具有较好的效果，这两个数据集合必须具有相似的规则性，相对熵 $\mathrm{relEntropy}(p|q)=\sum_{x\in C_X}p(x)\log\frac{p(x)}{q(x)}$ 用来比较两个数据集合的规则性，这个值越小说明两个数据集合越相似，当 $p=q$，相对条件熵为 0，说明这两个数据集合完全一致。同样，对异常检测来讲，相对条件熵 $\mathrm{relCondEntropy}(p\mid q)=\sum_{x,y\in C_x,C_y}p(x,y)\log\frac{p(x\mid y)}{q(x\mid y)}$ 的值越小越好。

属性 A 在数据集合 X 中的信息增益 $\mathrm{Gain}(X,A)=H(X)-\sum_{v\in \mathrm{Value}(A)}H(x_v)\log\frac{|x_v|}{|X|}$，其中 $\mathrm{Value}(A)$ 是 A 的所有可能值的集合，x_v 是当属性 A 具有值 v 时的 X 的一个子集。

入侵检测可以看作是一个分类问题，也就是将一个事件归为正常或者异常。每个事件都具有许多特征属性(features)，对于某个特征属性来说，信息增益越高，说明这项特征属性对数据记录的影响越大。为了保证分类模型的准确性，在构建分类模型时应该尽量使用信息增益高的特征属性。

15.5 注记

采用数据挖掘等方法对安全审计数据进行分析以达到入侵检测的目的，是目前的研究热点，已经有很多相关的研究成果。在这方面，也进行了一些技术研究和尝试。采取的技术路线是：使用数据挖掘中的数据分类、关联分析和序列挖掘，对基于网络和基于主机的安全审计数据进行智能化的分析处理，通过提取数据本身存在的规律性，帮助系统生成入侵检测规则及建立异常检测模型，最大限度地降低在处理安全审计数据时对先验知识的要求。基于网络的审计数据主要通过 tcpdump (tcpdump 是一种基于 Berkeley Packet Filter(BPF)的网络数据流分析工具)获取；基于主机的审计数据则通过 Solaris 操作系统的 BSM 模块获取特权进程的系统调用。

针对不同的安全审计数据源，建立包括数据获取、数据预处理、数据挖掘、结果分析等各个模块的完整的处理单元。

根据国内外研究人员的已有成果，结合所做的技术尝试，我们认为将数据挖掘技术应用于入侵检测在理论上是可行的，在技术上要建立这样一套系统也是可能的，其技术难点主要在于如何根据具体应用的要求，从关于安全的先验知识出发，提取出可以有效反映系统特性的特征属性(features)，应用合适的算法进行数据挖掘。技术难点还存在于结果的可视化，以及如何将挖掘结果自动地应用到实际的入侵检测系统中。

目前国际上这个方向的研究非常活跃，而且这些研究工作多数得到了美国国防部高级研究计划署(DARPA)、国家自然科学基金(NSF)等官方机构的支持。但是我们也发现，目前对于将数据挖掘技术运用于入侵检测方面的工作，总体而言还停留在理论研究的阶段，对于实际应用尚需要进一步的努力。

有关入侵检测和数据挖掘的技术进展，读者可重点关注以下两个网站：

- http://www.raid-symposium.org/，该网站主要介绍入侵检测技术的发展动态；
- http://www.sigkdd.org/，该网站主要介绍数据挖掘技术的发展动态。

参考文献

[1] Rakesh Agrawal, Ramakrishnan Srikant. Fast algorithms for mining association rules. In Proc. of the 20th Int'l Conference on Very Large Databases, Santiago, Chile, September 1994

[2] Chen M, Han J, Yu P. Data mining: An overview from database perspective. IEEE Transactions on Knowledge and Data Eng., 8(6):866-883, December 1996

[3] Jiawei Han, Micheline Kamber. Data Mining: Concepts and Techniques. Morgan Kaufmann Publishers, 2000

[4] Wenke Lee. A Data Mining Framework for Constructing Features and Models for Intrusion Detection Systems, PhD thesis, Columbia University, 1999

[5] Lee W, Stolfo S, Chan P, Eskin E, Fan W, Miller M, Hershkop S, Zhang J. Real time data mining-based intrusion detection. Proc. Second DARPA Information Survivability Conference and Exposition, pp. 185-100, 2001

[6] Leonid Portnoy, Eleazar Eskin and Salvatore J. Stolfo. Intrusion detection with unlabeled data using clustering. In Proceedings of ACM CSS Workshop on Data Mining Applied to Security (DMSA-2001). Philadelphia, PA: November 5-8, 2001

[7] Ghosh A K, Wanken J, Charron F. Detecting anomalous and unknown intrusions against programs. In Proceedings of the 1998 Annual Computer Security Applications Conference (ACSAC98), Dec 1998

[8] Hofmeyr S A, Forrest S, Somayaji A. Intrusion detection using sequences of system calls. Journal of Computer Security, 6: 151-180, 1998

[9] Wenke Lee, Dong Xiang. Information-Theoretic measures for anomaly detection. In the 2001 IEEE Symposium on Security and Privacy, Oakland, CA, May 2001

第 16 章　软件安全性分析方法与技术

随着信息技术的发展和应用，软件在信息系统中的核心地位越来越突出。软件系统的安全性不仅涉及其各类安全功能和安全机制的设计，而且涉及软件的具体实现方式。目前无论是纯粹的软件系统还是硬件系统中的软件代码，都出现了各种各样的漏洞和安全问题，严重影响了软件系统中安全功能和安全机制的有效性。软件安全性分析是对软件实现的安全性进行评估的一项重要工作，也是软件安全测评等工作的基础。本章将介绍以下几种软件安全性分析方法与技术。

(1) 程序切片。主要根据用户指定的切片准则提取相关的代码，去除不相关的代码，可减少复杂代码的干扰，降低分析难度，帮助分析人员更清晰地分析其关注的问题。目前主要有过程内切片、过程间切片、动态切片、条件切片等方式，根据不同的业务需求采用不同的切片方式。

(2) 模型检验。主要利用形式化方法分析相关系统是否具备相应的属性，它通过形式化方式对目标系统进行描述，利用逻辑表达式表示分析人员所关注的属性，然后分析该系统是否满足该表达式。目前该方法已广泛应用于软硬件系统设计、协议设计等领域，已能实现对 10^{120} 状态规模的信息系统的分析。

(3) 动态污点传播。主要用于分析特定数据在程序中的使用过程，辅助用户分析该数据内容对程序行为的影响。该方法目前广泛应用于软件漏洞分析、恶意代码分析等方面，已有一批实验系统发布。

上述方法目前在软件安全性分析中已广泛应用，但不仅仅用于软件安全性分析。本章主要介绍上述各种方法的一些基本概念、基本原理和典型应用。

16.1　程序切片

程序切片(program slicing)是一种用于分解程序的程序分析技术，在软件安全性分析、软件行为分析等方面具有广泛应用。程序切片是 1979 年 M. Weiser 博士在其博士论文中首次提出的，后来在其发表的论文中进行了进一步完善和推广[1]。

程序切片采用通俗的说法，一般有两种描述方法[2]：

(1) 对于程序中指令 p 处变量 x 的切片是指程序中所有可能影响变量 x 在指令 p 处的取值的语句。

(2) 对于程序中指令 p 处变量 x 的切片是指对程序进行删减，确保所有在指令 p 处 x 的取值序列不变的情况下所得到的程序。也就是说，所删减后得到的程序和原程序的行为，对于指令 p 处的变量 x 而言是一致的。

程序切片按照切片分析目标的不同可以分为过程内切片和过程间切片。过程内切片是指对只有一个过程组成的程序进行切片，其分析目标是单一的过程，无需考虑

其他过程间的调用关系。相反，过程间切片则是对由多个过程组成的程序进行切片，其分析过程更复杂。和过程内切片相比，过程间切片更贴近实际需求。下面将重点从过程内切片和过程间切片两方面来介绍程序切片技术，然后简要介绍一下其他切片技术的发展情况。

16.1.1 过程内切片

在讨论过程内切片之前，首先引入几个定义，并简要介绍 M. Weiser 最早提出的基于数据流方程切片方法的核心思想，随后将重点介绍基于依赖图的切片方法。

定义 16.1.1 **状态轨迹**是程序执行的轨迹，表示每条语句执行之前对所有变量值的快照。

状态轨迹是用于描述程序状态的一种方法，通过所有变量的值来描述程序的不同状态，不同状态的序列即状态轨迹的序列就构成了程序的执行流程。

定义 16.1.2（切片准则） 程序 P 的切片准则是一个二元组 $<n,V>$，其中 n 是程序中的一条语句，V 是 P 中变量的一个子集。

切片准则 $C=<n,V>$ 定义了一个投影函数 Proj_C。

定义 16.1.3 令 $T=(t_1,t_2,\cdots,t_m)$ 是一个状态轨迹，$\forall m\in N$。s 是从变量名到值的函数，则有

$$\mathrm{Proj}'_{<n,V>}(m,s)=\begin{cases}\varnothing & m\neq n\\ (m,s\mid V) & m=n\end{cases}$$

其中，$s|V$ 表示 s 被限制在域 V 的范围内；$\varnothing$ 是一个空的字符串。现在把 Proj' 扩展到整个的轨迹：$\mathrm{Proj}_{<n,V>}(T)=\mathrm{Proj}'_{<n,V>}(t_1)\cdots\mathrm{Proj}'_{<n,V>}(t_m)$。

定义 16.1.4（切片） 程序 P 的一个关于切片准则 $C=<n,V>$ 的切片 S 是具有下列特性的任何可执行程序：

（1）S 通过从 P 中删除零条或多条语句获得；

（2）无论何时，只要程序 P 在输入 I 和状态轨迹 T 处停止，则 S 也会在输入 I 和状态轨迹 T' 处停止，并且 $\mathrm{Proj}_C(T)=\mathrm{Proj}'_C(T')$，其中 $C'=<\mathrm{SUCC}(n),V>$，$\mathrm{SUCC}(n)$ 是 n 在源程序 P 中的最近后继，它不在切片中；或者如果 n 也在切片中，$\mathrm{SUCC}(n)$ 就是 n 本身。

由上述定义可知，程序切片具有下面两个性质：

（1）程序切片必须通过对源程序进行语句删除而得到，切片所具有的程序语句不能超出原有程序范围；

（2）通过对程序切片准则窗口所观察到的切片行为必须与源程序的相应行为等价。

M. Weiser 提出了一种基于数据流方程的静态切片算法，这种方法通过迭代计算 CFG 中每个节点相关变量的集合来获得程序切片。给定切片准则 $C=<n,V>$，M. Weiser 方法的计算步骤如下：

（1）进行第一次循环，计算直接相关变量和语句。

① 对 CFG 中某个节点 s，其直接相关变量集合 $R_C^0(s)$ 定义为：C 中 n 的直接相

关变量集合为 V；对 CFG 中的其他节点 s，它的直接相关变量的集定义为

$$R_C^0(s) = \bigcup_{\mathrm{Pred}(s,t)} \{v \mid v \in R_C^0(t) \wedge v \notin \mathrm{Def}(s)\} \cup \{v \mid v \in \mathrm{Use}(s) \wedge \mathrm{Def}(s) \cap R_C^0(t) = \varnothing\}$$

② 直接相关语句集可根据直接相关变量来获得，记为

$$S_C^0 = \{s \mid \exists t, \mathrm{Pred}(s,t) \wedge (\mathrm{Def}(s) \cap R_C^0(t) = \varnothing)\}$$

（2）进行迭代，计算间接相关变量和间接相关语句。

① 间接相关变量集合记为 $R_C^k(s)$，其计算过程如下：

$$R_C^{k+1}(s) = R_C^k(s) \cup \left(\bigcup_{b \in B_C^k} R^0_{(b,\mathrm{Use}(b))}(s)\right)$$

其中 $B_C^k = \{b \mid \exists s, s \in S_C^k, b$ 的执行决定 s 是否执行$\}$是控制 S_C^k 中一个语句的所有控制节点的集合。

② 把控制节点加到 S_C^k 中，进一步计算间接相关语句：

$$S_C^{k+1} = B_C^k \cup \{s \mid \exists t, \mathrm{Pred}(s,t) \wedge (\mathrm{Def}(s) \cap R_C^{k+1}(t) \neq \varnothing)\}$$

（3）重复第二步直到 S 不再增大为止。

M. Weiser 的程序切片算法只能处理 3 种基本的结构，即顺序、选择和循环，对模块化程序中出现的过程和过程调用无能为力，其详细的计算过程可参考文献[1]和[2]。在此基础上，S. Horwitz 等人提出了基于依赖图的过程内切片方法。下面重点介绍该方法的原理和具体的切片算法。

定义 16.1.5（过程内切片） P 是只包含一个过程的程序，切片准则是$<n,V>$，则程序 P 关于此准则的程序切片是由程序 P 中那些影响 V 中变量在兴趣点 n 的值的所有语句和控制谓词组成的集合，这种切片称为过程内切片，又称为**静态后向切片（static backward slice）**。

定义 16.1.6 如果程序 P 关于切片准则$<n,V>$的切片是由程序 P 中所有受到兴趣点 n 的变量集 V 中变量值影响的语句和控制谓词构成的集合组成，则该切片被称为**静态前向切片（static forward slice）**。

程序依赖图（program dependence graph）是一种用于描述控制依赖、数据依赖等依赖关系的图，记为 G_P。其中的节点集合是程序中的每一条指令和一些附加的特殊节点组成。特殊的节点主要包括：

（1）程序的入口节点，记为 Entry。

（2）对于每一个变量，在未初始化就使用时，需增加一个初始化节点，记为“x:= InitialState(x)”。

（3）在程序结束时语句中的变量，都需增加一个最后使用节点，记为“FinalUse(x)”，表示访问该变量的最后计算结果。

边的集合则由控制依赖边和数据依赖边组成。控制依赖边的起点要么是程序的入口节点，要么就是判断条件语句。每个控制依赖边会标示为“T”或者“F”，以表示判断条件的“真”、“假”。如果 v_1 和 v_2 之间存在控制依赖关系，记为 $v_1 \rightarrow_c v_2$，表示当程序执行时，如果程序会最后终止，并且判断条件语句 v_1 的评估结果和依赖关系边

上的标示一致，则节点 v_2 中的语句一定会执行。

当且仅当满足以下条件时，程序依赖图 G_P 中节点 v_1 和 v_2 之间存在一条控制依赖边：

(1) v_1 是入口节点，v_2 是程序中的一条指令，且不控制依赖于任何一条循环或其他条件判断语句。则存在 $v_1 \rightarrow_c v_2$，该边标示为"T"。

(2) v_1 是一个控制条件语句，当 v_1 是循环的控制条件语句时，如果 v_2 的指令在该循环体内，则存在 $v_1 \rightarrow_c v_2$，并且标示为"T"；如果 v_1 是普通条件判断语句，v_2 是根据该条件判定结果确定是否执行的语句，则存在 $v_1 \rightarrow_c v_2$，并且根据判断条件的真假，相应的标示为"T"或"F"。

节点 v_1 和 v_2 之间存在一条数据依赖边，意味着当程序中节点 v_1 和 v_2 对应的指令顺序发生改变时，程序的运算结果将发生改变。数据依赖边主要分为**数据流依赖(flow dependence)**和**定义顺序依赖(def-order dependence)**。

当且仅当以下条件满足时，节点 v_1 和 v_2 之间存在数据流依赖关系：

(1) 节点 v_1 定义了变量 x；

(2) 节点 v_2 使用了变量 x；

(3) 存在一条执行路径可从节点 v_1 到达节点 v_2，并且，在该路径中，节点 v_2 使用变量 x 之前，不存在其他指令对 x 进行重新定义。所有变量的最初初始化均紧接程序入口节点之后，变量的最后使用在紧邻程序结束之前。

节点 v_1 和 v_2 之间的数据流依赖关系记为 $v_1 \rightarrow_f v_2$。

数据流依赖又可分为**循环数据流依赖(loop carried flow dependence)**和**非循环数据流依赖(loop independent flow dependence)**。若节点 v_1 和 v_2 满足以上 3 个条件的同时，满足以下条件：

(4) 满足(3)的执行路径包含一条边回到循环 L 的条件判断语句；

(5) 节点 v_1 和 v_2 均包含在循环体内。

则称节点 v_1 和 v_2 之间存在循环数据流依赖，记为 $v_1 \rightarrow_{lc(L)} v_2$。

非循环数据流依赖则是在满足条件(1)、(2)和(3)的同时，存在一条路径满足(3)但不存在边回到包含节点 v_1 和 v_2 循环 L 的条件判断语句，记为 $v_1 \rightarrow_{li} v_2$。两个节点 v_1 和 v_2，可能会同时存在 $v_1 \rightarrow_{lc(L)} v_2$ 和 $v_1 \rightarrow_{li} v_2$。

节点 v_2 定义顺序依赖于节点 v_1 当且仅当以下条件满足：

(1) 节点 v_1 和 v_2 都定义了同一个变量 x；

(2) 节点 v_1 和 v_2 对于任意同时包含两个节点的判断条件语句，它们都在同一个分支；

(3) 存在一个节点 v_3，使得 $v_1 \rightarrow_f v_3$ 和 $v_2 \rightarrow_f v_3$；

(4) 在程序的抽象语法树(abstract syntax tree)中，节点 v_1 在 v_2 的左边。

这样，节点 v_3 表明从节点 v_1 到 v_2 存在**定义顺序依赖**，记为 $v_1 \rightarrow_{do(v_3)} v_2$。

对于程序依赖图 G 中节点 s 的切片(记为 G/s)，是 G 的一个子图，该子图的节点集合包含了所有 s 控制依赖或数据依赖的节点；该子图的边是所有和 s 相关的控制依赖和数据流依赖边，以及部分定义顺序依赖边。其中，对于定义顺序依赖边

$v \rightarrow_{do(u)} w$，当且仅当节点 u 是该子图节点，则该定义顺序边属于该子图。

对于子图点的集合可定义如下：

$$V(G/s) = \{w \mid w \in V(G) w \rightarrow^{*}_{c,f} s\}$$

对于单一节点 s 的定义，还可将其扩展到节点集合 $S = \bigcup_i s_i$ 的定义，其定义如下：

$$V(G/S) = V\left(G/\left(\bigcup_i s_i\right)\right) = \bigcup_i V(G/s_i)$$

对于子图边的定义如下：

$$\begin{aligned} E(G/S) = &\{(v \rightarrow_f w) \mid (v \rightarrow_f w) \in E(G) \quad v,w \in V(G/S)\} \\ &\cup \{(v \rightarrow_c w) \mid (v \rightarrow_c w) \in E(G) \quad v,w \in V(G/S)\} \\ &\cup \{(v \rightarrow_{do(u)} w) \mid (v \rightarrow_{do(u)} w) \in E(G) \quad v,w,u \in V(G/S)\} \end{aligned}$$

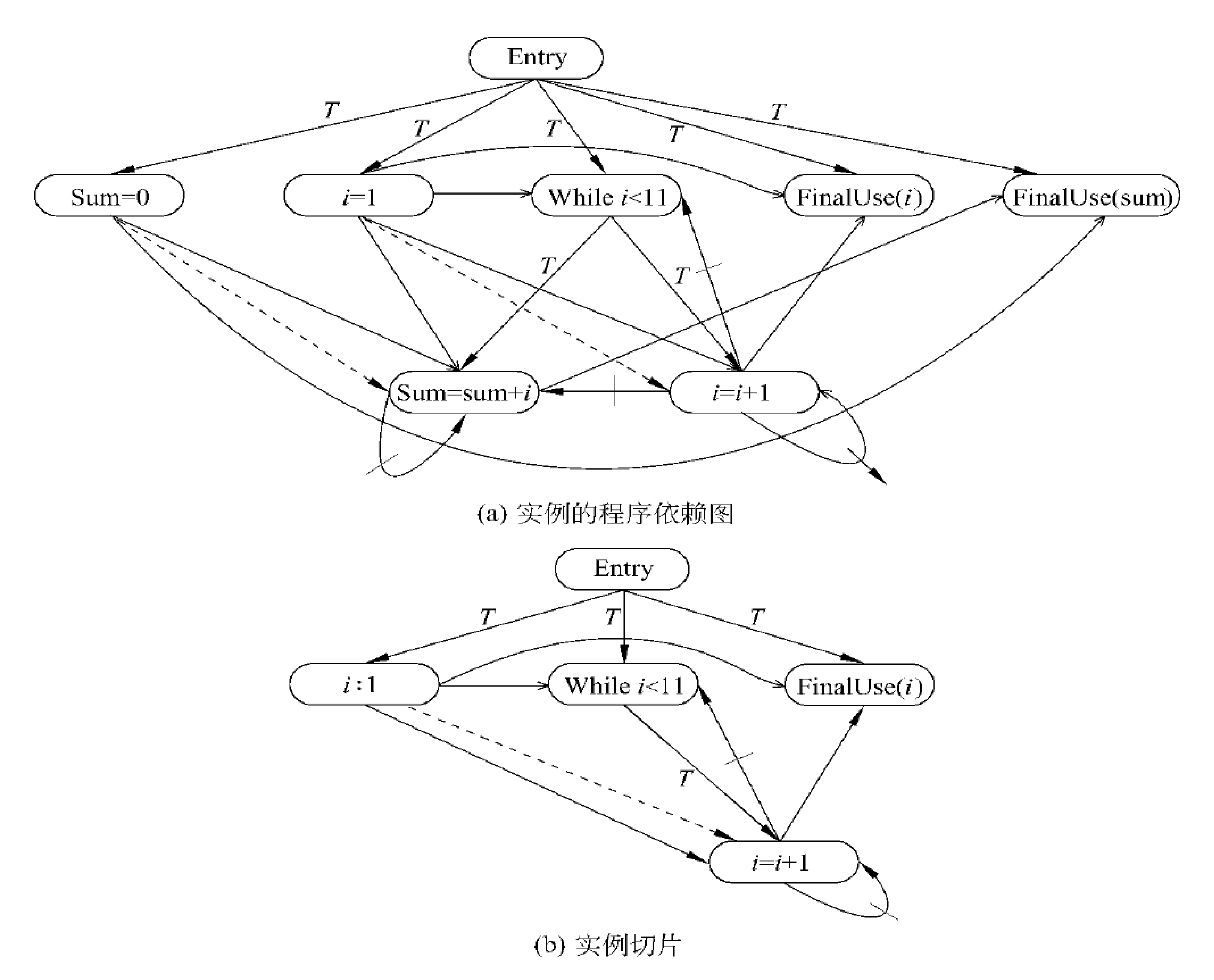
从以上的定义可以看出，在过程内切片中，最重要的就是计算该子图的节点集合。下面是节点集合计算的算法：

```
procedure Mark VerticesOfSlice(G,S)
declare
   G: 程序依赖图
   S: G中的一个节点集合,切片的准则
   WorkList: 一个节点集合
   v,w: G中的节点
begin
   WorkList:=S
   While WorkList 不为空 do
       从 WorkList 中选择一个节点 v,从 WorkList 中删除;
       标记 v;
       for 对于没有标记的节点 w,并且 E(G)中存在 w→f v 或者 w→c v do
           将 w 添加到 WorkList 集合
       od
   od
end
```

通过该算法，标示的节点即为根据节点集合 S 切片所得子图的节点集合。下面是一个具体实例，左边为源程序，右边为切片的结果，根据的切片准则是<FinalUse(i)，i>。

例 16.1.1 实例如图 16.1 所示。

```
Program Main
    sum:=0;
    i:=1
    while i<11 do
        sum:=sum+I;
        i:=i+1;
    od
end(sum,i)
```



(a) 实例的程序依赖图

(b) 实例切片

图 16.1　过程内切片实例

切片结果：

```
Program Main
    i:= 1
    while i<11 do
        i:=i+1;
    od
end(i)
```

16.1.2　过程间切片

随着软件规模的扩大，单一过程已不能满足软件功能实现的需要，这样，多个过程的程序切片的问题也越来越重要。早在 1981 年 M. Weiser 就给出了一个过程间切片的算法，他的基本思想是将原有的切片准则进行扩展，对被调过程(called procedure)和施调过程(call procedure)添加新的切片准则，程序切片最终变成一组切片准则的切片。该算法由于调用上下文问题而不够精确，包含了较多不必要的语句。当计算从 P 调用的子程序 Q 的切片时，也将对所有调用 Q 的过程进行切片。就如同程序流从 P 执行到 Q 时，又从 Q 执行到 P'。

1990 年，3 位美国学者 S. Horwitz、T. Reps 和 D. Binkley 在总结前人经验的基础上，利用依赖图解决了含有多个过程程序的过程间切片问题[3]。

S. Horwitz 等人引入系统依赖图(SDG)的概念来表示具有多个过程的程序的依赖图。一个 SDG 图是由一组过程间控制依赖边和流依赖边连接起来的过程依赖图(PrDG)组成，PrDG 类似于 PDG，但它还包括表示由于调用而形成的调用语句、参数传递、可传递流依赖的节点和边。过程间切片和过程内切片相比较，所需要解决的主要两方面的问题：一个是参数传递过程表示；另一个是调用上下文的表示。

在系统依赖图中，当过程 P 调用过程 Q 时，参数和返回值均借助于中间变量传递。在调用时，P 首先将参数复制到中间变量，然后 Q 利用这些数值初始化其内部变量；在返回之前，Q 首先将返回值复制到中间变量，然后再传递给 P。

基于这样一个基本过程，引入了参数节点，以准确描述参数传递过程。对于调用过程 P，在调用 Q 的位置，需要增加两类节点 actual-in 和 actual-out。actual-in 节点用于描述传递给 Q 的参数，actual-out 用于描述 Q 返回的返回值。在被调用过程 Q 也要增加两类节点 formal-in 和 formal-out，formal-in 用于描述传递进来的参数，formal-out 用于描述 Q 返回的返回值。

另外，还需要增加以下边：①首先增加一条从过程 P 中的调用过程 Q 的节点到过程 Q 的入口节点的边，以描述调用关系，称之为调用边；②对于参数传递过程，需要增加对应的从过程 P 中的 actual-in 节点到过程 Q 中的 formal-in 节点的边，称之为参数传递边；③对于参数返回过程，需要增加对应的从过程 Q 中 formal-out 节点到 P 中 actual-out 节点的边，称之为返回值边。其中调用边是控制依赖边，参数传递边与返回值边属于数据依赖边。

在实际分析过程中，在过程 P 调用过程 Q 时，actual-in 节点以如下方式表示：

r_in:$=e$，其中 r 是过程 Q 的参数名称，e 是过程 P 传入的参数。

actual-out 节点以如下方式表示：

a:$=r$_out，其中 r 是过程 Q 中返回参数的名称，a 是过程 P 中保存该返回值的变量。

formal-in 节点以如下方式表示：

r:$=r$_in，其中 r 是过程 Q 的输入参数名称。

formal-out 节点以如下方式表示：

r_out:$=r$，其中 r 是过程 Q 的返回参数名称。

可传递的依赖边，又称 summary 边，加到 actual_in 节点和 actual_out 节点之间来表示由于调用过程而形成的可传递流依赖。如果在被调用过程中相应的 formal_in节点和 formal_out 节点之间存在一条控制路径、流边或 summary 边，则在 actual_in 和 actual_out 之间加一条 summary 边。

使用 3 种类型边把过程依赖图(PrDG)连接起来形成 SDG。

(1) 调用边(call edge)：在每个调用位置节点和相应的过程入口节点之间架上调用边。

(2) 参数输入边(parameter_in)：在调用位置的每个 actual_in 节点到被调用过

程相应的 formal_in 节点之间增加参数输入边。

（3）参数输出边（parameter_out）：从被调用过程的 formal_out 节点加一条边到相应的调用位置的 actual_out 节点。

为解决调用上下文的问题还需要在该系统依赖图中引入表示由过程调用引起的可达数据依赖边，实际上也就是要找出过程的 actual_out 参数和 actual_in 参数间的相互依赖关系。这里先引入一个实例，其中 Main 是主程序，涉及的过程包括 A(x,y)、Add(a,b)、Increment(z)。图 16.2 所示是经过调用关系处理后的依赖图。

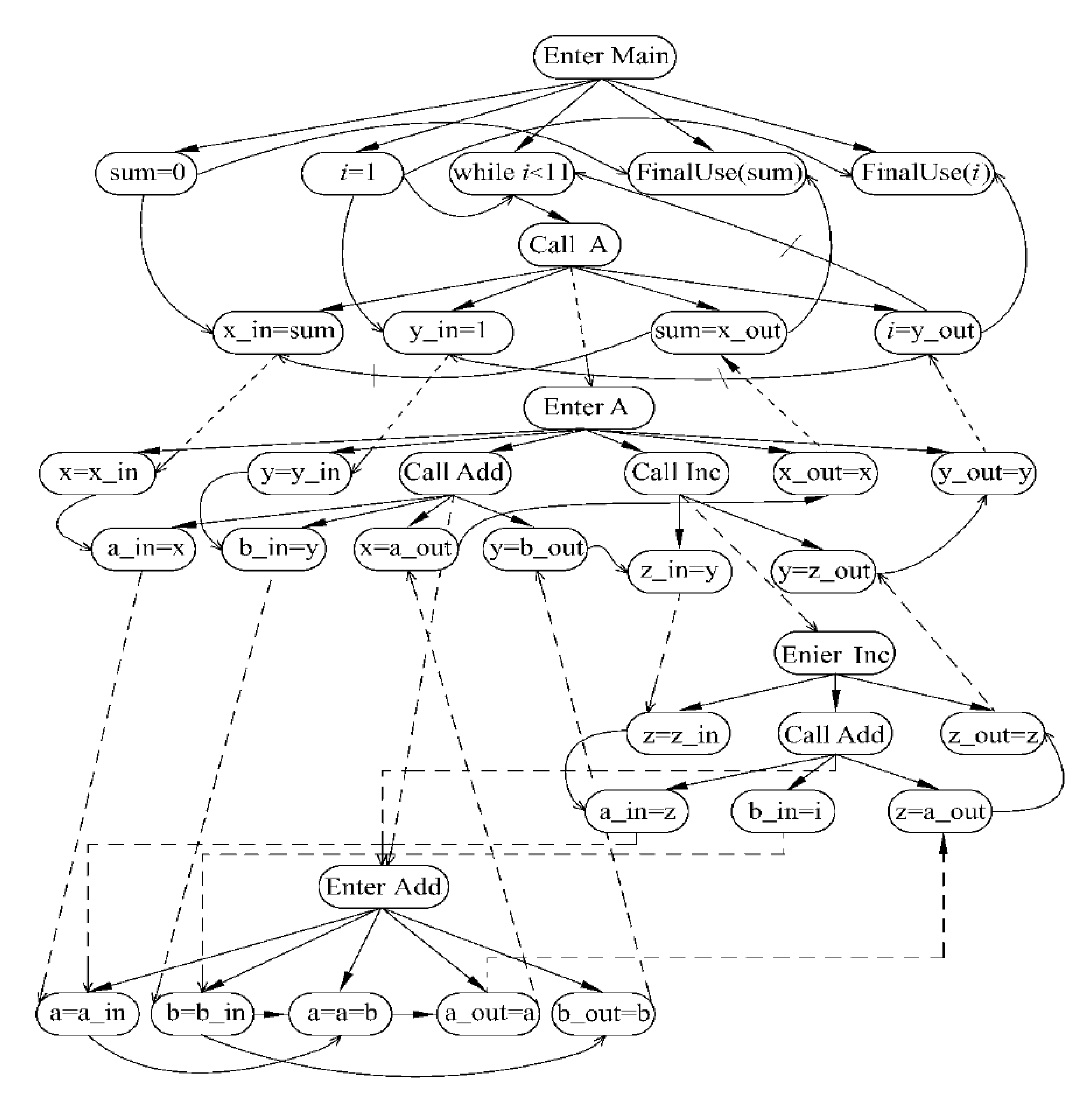


图 16.2　经过参数关系处理和调用关系处理后的依赖图

例 16.1.2：

```
Program Main
    Sum:=0;
    i:=1;
    while i<11 do
        call Add(sum,i);
    od
  end(sum,i)
```

```
Procedure Add(x,y)
  Call Add(x,y);
  Call Increment(y);
Return
Procedure Add(a,b)
  A:=a+b;
Return
Procedure Increment(z)
    Call Add(z,1)
return
```

以上是对参数传递与返回值描述的方法，另外一个问题是对调用上下文关系的进一步准确描述。连接文法主要用于建立过程调用的结构模型，是一种通过对上下文无关文法进行扩充而来。它在上下文无关文法的基础上，对终结符和非终结符增加了一些属性，并通过属性等式的方式对这些属性进行定义。对于每一个产生式如下列形式：

$$p: X_0 \to X_1, X_2, \cdots, X_k$$

其中 X_i 表示文法符号。

每个文法符号的出现表示相对应的属性存在，每个产生式有一系列的属性等式利用属性函数的方式描述和定义属性。一个符号 X 的属性被分为综合属性和集成属性两个集合。每一个产生式 p 的属性依赖关系可以通过依赖图来描述，记为 $D(p)$。该依赖图表示如下：

（1）对于每一个属性 b，该图有一个节点 b'；

（2）如果属性 b 在属性等式的右边定义了属性 c，则该图存在一条边 $b' \to c'$。

对于一个程序，按照以下方式建立一套文法：

（1）对于程序中的每一个过程 P，设置一个非终结符 P；

（2）对于每一个过程 P，建立一个产生式 p：$P \to \beta$，如果过程 P 未调用任何过程，则 $\beta = \varepsilon$，即过程 P 的产生式为 p：$P \to \varepsilon$，否则 β 是 P 调用过程所对应的非终结符的序列，比如过程 P 调用了过程 Q 和 N，则其对应产生式为 p：$P \to QN$；

（3）对于非终结符 P，如果节点 P 存在一个 actual-in 节点，则其存在一个对应的继承属性；如果节点 P 存在一个 actual-out 节点，则其存在一个对应的综合属性。非终结符 P 的属性 a 标示为 $P.a$。

根据例 16.1.2 可以得到以下产生式：

$$\text{Main} \to A \quad A \to \text{Add Increment} \quad \text{Add} \to \varepsilon \quad \text{Increment} \to \text{Add}$$

其中 Main 过程调用了过程 A；过程 A 调用了过程 Add 和 Increment；过程 Add 没有调用任何其他过程，过程 Increment 调用了过程 Add。

连接文法中引入属性主要是为了描述过程内部参数节点和返回值节点之间的依赖关系。这些依赖关系的分析主要借助于前面介绍的过程内的切片方法。为了描述这些属性之间的依赖关系，连接文法中引入了属性等式。属性等式的描述如下：

对于表达式 p 中的属性 $P.a$，假设 v 是过程依赖图 G_P 中 $P.a$ 所对应的节点，可

以得到关于 $P.a$ 的属性等式 $P.a=f(\cdots,Q.b,\cdots)$，其中，$Q.b$ 是 G_P/v 中 p 所对应的参数（或返回值）节点。

对于实例中产生式 Main→A 所对应的属性等式如下所示：

$$A.x_in = f_1(A.x_out, A.y_out)$$
$$A.y_in = f_2(A.y_out)$$
$$A.x_out = f_3(A.y_out)$$
$$A.y_out = f_4(A.y_out)$$

图 16.3 是例 16.1.2 的属性依赖图。

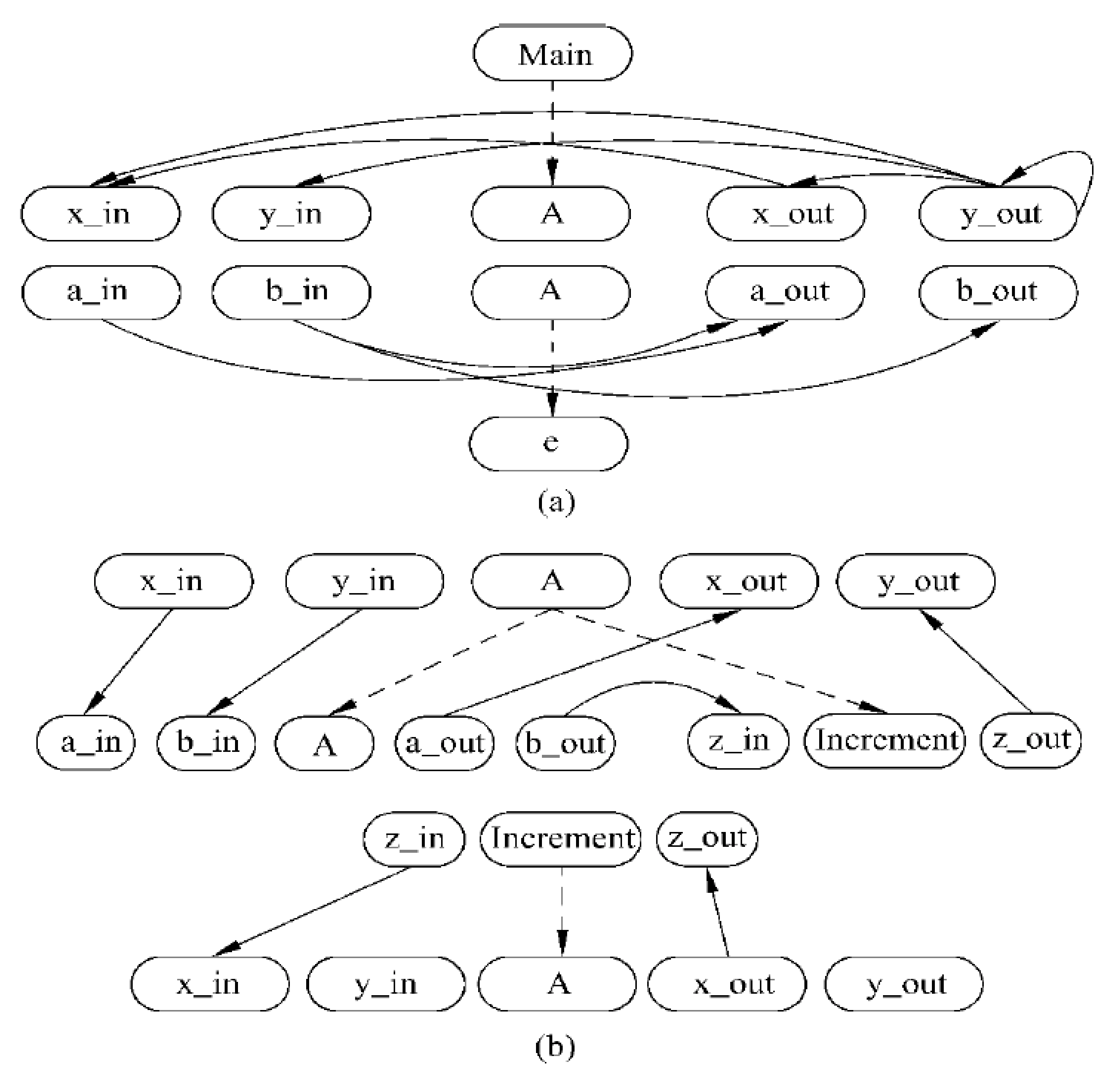


图 16.3　属性依赖图

为了描述属性间的传递依赖关系，引入了特征子图的概念。在定义特征子图之前，先介绍以下定义。

(1) 对于一个有向图 $G=(V,E)$，其中从节点 a 到节点 b 的路径定义为 $[v_1,v_2,\cdots,v_k]$，其中 $a=v_1$，$b=v_k$，且 $\{(v_i,v_{i+1})\mid i=1,2,\cdots,k-1\}\subseteq E$。

(2) 对于一个有向图 $G=(V,E)$ 和一个节点集合 $V'\subseteq V$，G 在 V' 上的映射 $G//V'=(V',E')$，其中

$$E'=\{(v,w)\mid v,w\in V'，且 G 中存在路径[v=v_1,v_2,\cdots,v_k=w],v_2,\cdots,v_{k-1}\notin V'\}$$

假设 r 是属性树 T 中的一个节点，r 节点的属性集合定义为 $A(r)$，以 r 为根节点的属性子树，记为 T_r，则对于非终结符 r 的特征子图定义为 $r.C=D(T_r)//A(r)$。

非终结符 X 的特征子图是 TDS(X)，初始时 TDS(X)为空。TDS(X)中的边实际上就是 X 对应的过程的各参数间的数据依赖边。辅助图 TDP(P)描述了产生式

中非终结符的出现情况之间的关系，也就是产生式中每次调用的过程的参数间的依赖关系。它可以是不同过程的参数间的依赖，也可以是同一过程的不同次出现的参数间的依赖关系。对形如 Add→ε 的产生式，其 TDP 与 TDS 相同，都是 Add 的参数间的依赖，对形如 A→Add|Increment 的产生式，TDP 是 A、Add、Increment 参数间的依赖关系，而 TDS 则仅是 A 的参数间的依赖。算法中的一个基本操作是 AddEdgeAndInduce(TDP(p)，(a,b))，其中 p 是个产生式，(a,b)是一对存在依赖关系的属性出现。该操作执行以下 3 个动作：①将(a,b)加入到辅助图 TDP(p)中；②加入使 TDP(p)可达封闭的额外边到 TDP(p)中，也就是加入因(a,b)而可到达的新的可达边；③对 TDP(p)中形如($X_0.m$,$X_0.n$)的边，X_0 是产生式中非终结符 X 的左边出现，则将($X.m$,$X.n$)加入到 TDS(X)中。

构造特征子图算法如下：

```
procedure ConstructSubCGraphs(L)
declare
  L: 连接文法
  p: L中的一个产生式
  Xi,Xj,X̂: L中的非终结符的出现
  a,b: L中非终结符的属性
  X: L中的非终结符
begin
/* 初始化 TDS 图和 TDP 图 */
   for L中的每一个 X do
       TDS(X):=为 X的每一个属性 X.b生成一个节点,但不生成任何边。
   od
   for L中的每一个产生式 p do
       TDP(p) :=为每一个属性 Xj.b生成一个节点,但不生成任何边。
       for 对于 p中的每一个属性 Xj.b do
          for 定义 Xj.b的属性等式中的每一个参数 Xi.a do
              增加一条边(Xi.a,Xj.b)到图 TDP(p)中。
              假设非终结符 X对应于 Xj
                  if i=0,j=0,且(X.a,X.b)∉TDS(X),then
                   在 TDS(X)中插入一条非标记边(X.a,X.b)
                  fi
                od
              od
         od
  /* 确定传递关系 */
     While 所有 TDS 图中存在一条非标记边(X.a,X.b) do
         mark(X.a,X.b)
        for 对于任意一个产生式 p中非终结符 X的每一次出现 X̂ do
              if (X̂.a,X̂.b)∉TDP(p) then
```

```
                    AddEdgeAndInduce(TDP(p),(X̂.a,X̂.b))
                fi
            od
        od
    end
```

图 16.4 所示是实例 16.1.2 最终的系统依赖图。

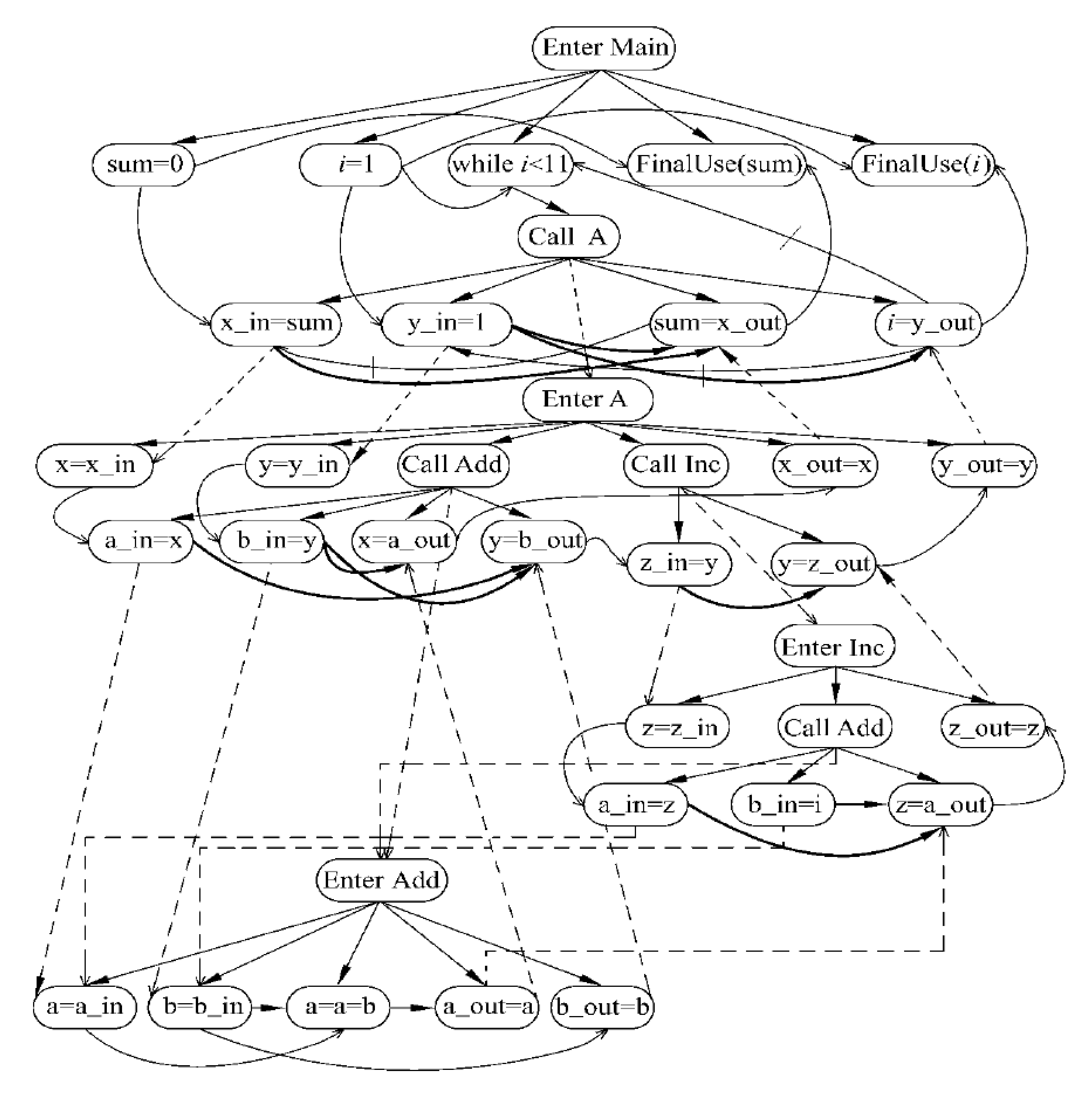


图 16.4　实例的系统依赖图

其中，AddEdgeAndInduce(TDP(p),(a,b))主要完成以下功能：

(1) 在 TDP(p)中增加边(a,b)；

(2) 补充需要的边，以确保 TDP(p)传递闭包；

(3) 对于(1)、(2)中增加的任意以下形式的边($X_0.m$,$X_0.n$)，X_0 表示非终结符 X 在属性等式 p 的左边，并且($X.m$,$X.n$)$\notin$TDS(X)，则($X.m$,$X.n$)增加到图 TDS(X)中。

下面是基于系统依赖图的过程间切片算法，图 16.5 所示是最终的切片结果。

基于系统依赖图的过程间切片算法如下。

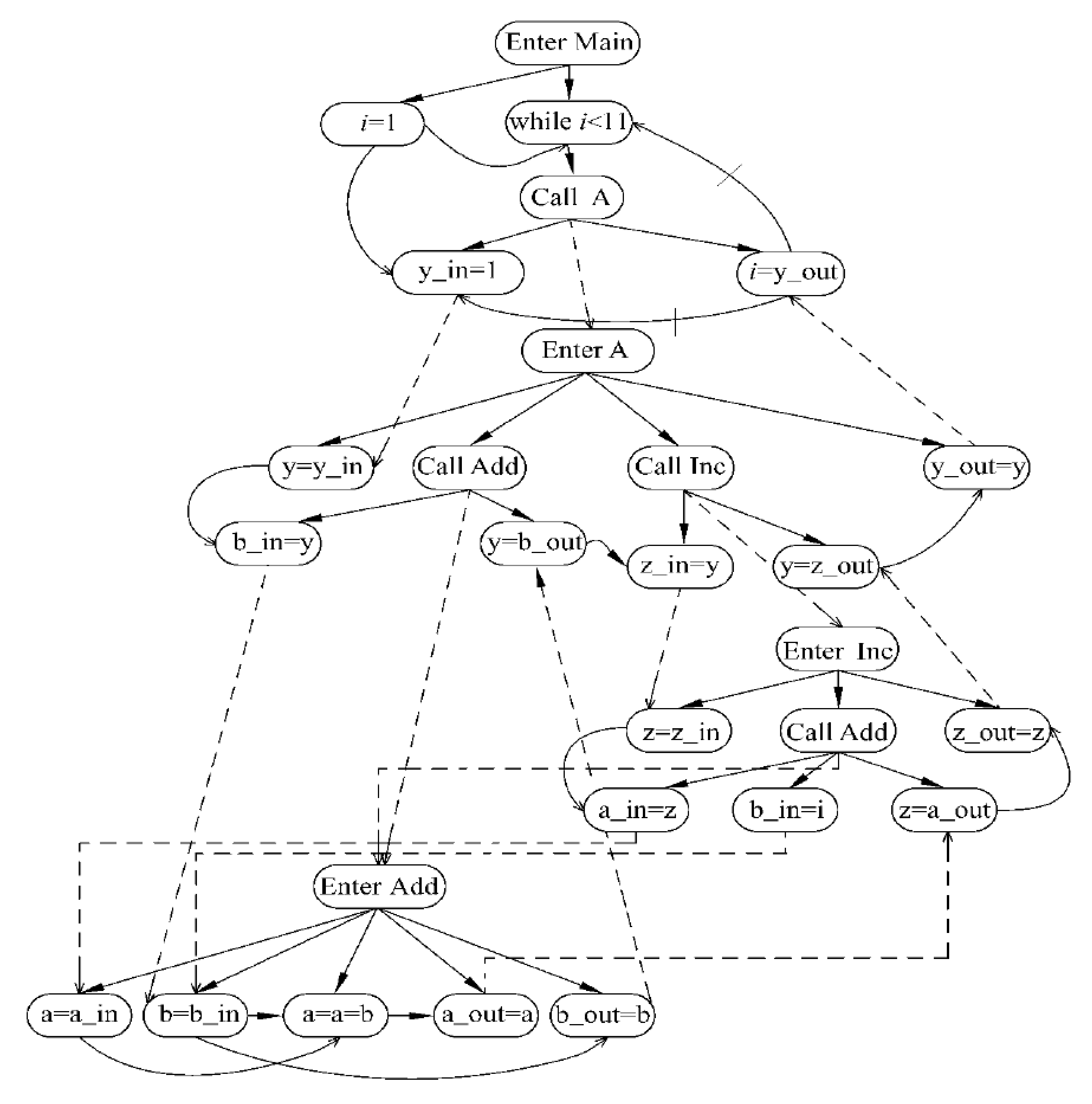


图 16.5 最终的切片结果

```
procedure MarkVerticesOf Slice(G,S)
declare
   G: 系统依赖图
   S,S′: G中的节点
begin
   /* 在不深入调用过程内部的情况下切片 */
    MarkReachingVertices(G,S,{def-order,parameter-out})
/* 对被调用的过程内部切片,不扩展到调用节点 */
    S′:=G中所有标记的节点
    MarkReachingVertices(G,S′,{def-order,parameter-in,call})
end
procedure MarkReachingVertices(G,V,kinds)
declare
   G: 系统依赖图
   V: G中的节点集合
   kinds: 边的种类
```

```
  v,w: G的节点
  WorkList: G中的节点集合
begin
  WorkList:= V
  while WorkList≠∅ do
    选择一个节点 v,并将其从 WorkList 集合中删除
    mark(v)
    for 每一个未标记节点 w,如果存在边 w→v,并且该边不属于 Kinds 中的任何一类 do
      将 w增加到 WorkList 中
    od
  od
end
```

基于系统依赖图的过程间切片仍是目前最经典的过程间切片算法之一，更详细的算法细节，读者可以参考文献[3]。本文中的相关实例也均来自于该文献。

16.1.3　其他切片方法

自从 M. Weiser 提出切片思想以来，针对不同的应用需要，切片技术也在不断发展和完善，随后又提出了动态切片(dynamic slice)[4]和条件切片(conditional slice)[5]等技术。其中动态切片由 B. Korel 和 J. Laski 最早提出，它与程序的某个特定输入 I_0 有关。动态切片准则是一个三元组 $<n,V,I_0>$，要计算程序 P 的动态切片就是要计算程序 P 中在某个特定输入 I_0 的情况下所有影响变量集合 V 中变量在 n 点的值的语句和谓词组成的集合。

有条件切片最早是由 G. Canfora、A. Cimitile 和 A. D. Lucia 这 3 位学者于 1998 年在文献[5]中提出的。该论文首次建立了如何计算条件切片的思想。从条件切片的定义来看，它是介于静态切片和动态切片之间的一种切片形式：进行条件切片时，假设满足某个谓词表达式的所有可能输入是 I_c，静态切片时程序的所有可能输入是 I，动态切片时考虑的某个特定输入是 I_0，则三者的关系是

$$I_c \subseteq I, \quad I_0 \in I$$

从中可以发现，三者的关系主要体现在切片准则的变化。如果切片准则是 $<n,V,I_0>$，则所得的结果就是动态切片；如果切片准则是 $<n,V,I_c>$，则所得的结果是条件切片；如果切片准则是 $<n,V,I>$，则就是静态切片。

16.2　模型检验

模型检验主要用于验证反应系统(reactive system)属性的正确性，特别是对于一些与安全息息相关的关键性系统。该方法已成功应用于发现复杂的工业产品中的细微错误，如电路设计、通信协议、数字控制器、软件核心模块等。除了静态分析、测试等传统的质量控制措施，模型校验将成为设计反应系统的标准程序之一。模型检验最初是由 Edmund M. Clarke 于 1981 年提出的，他也因为在这方面的卓越贡献而

获得图灵奖[6]。

反应系统是由多个组件组成,它们可以彼此之间交互,也可以和系统环境之间进行交互。和**功能系统**(functional system)相比,反应系统的特点在于其时序属性(temporal properties)。一个**属性**(property)是在特定时间所期望的行为的集合。该系统满足该属性当且仅当系统的每次执行的行为都属于该集合。从逻辑的角度,该系统利用语意模型(semantical model)描述,属性则采用逻辑表达式(logical formula)描述。系统的正确性因此转变为确定在该模型中表达式是否正确。

为了能够进行这样的验证,首先需要一个模型语言(modelling language)用于描述系统,一个描述语言(specification language)用于建立属性表达式,以及执行验证过程的推理算式(deductive calculus)或算法(algorithm)。通常被校验的系统采用有限状态转换图来描述,属性则形式化为时序逻辑表达式(propositional temporal logic),确定状态图是否满足这些表达式则通过一个搜索过程完成。1981 年,Clarke 等人提出模型校验时,模型校验还只能处理上千个状态的小规模并发系统。但近年来,通过应用巧妙的数据结构和启发式的搜索过程,目前对系统处理的规模已大大增加。

模型校验成功的原因之一就在于它是一种自动的验证方法。交互式方法更通用,但难以使用。自动方法虽然具有一定的局限性,但还是可以接受的。模型校验,用户只需要提供系统的模型和需要证明属性的表达式即可。当该模型可以满足属性表达式时,验证工具将停止;当该模型不能满足表达式时,该工具将指出为什么没有满足。这些反例将有助于确定模型或系统中的错误。

由于采用完全自动的方法,模型校验需要遍历系统中所有可能的状态。因此,别的推理方法可以处理无限状态的问题,但模型校验只能处理有限的状态。对于完全自动的方法,最大的问题就是状态爆炸:如果系统中任意的状态都描述为 n 个状态位,这样就有 2^n 种可能状态。在目前,可处理的状态数量将近 10^6。决策二叉树被用于描述状态空间,采用这种方法,可处理规模超过 10^{100} 个状态的系统。正是由于类似技术的出现,现有的工具才可能处理现实中的复杂系统。许多的企业已利用模型检验来验证实际的产品设计,如 Intel、Motorola、ATT、Fujitsu 和 Siemens。

模型校验的 3 个步骤:建模(modeling),即为所要分析的目标系统建立相应模型;描述(specification),即通过形式化语言对目标系统模型和属性进行描述;验证(verification),即基于前面的描述,分析目标系统是否具备相应的属性。

下面将在 16.2.1 小节中介绍对于反应系统的建模方式 Kripke 结构,在 16.2.2 小节中将介绍如何利用计算树逻辑来描述系统,基于计算树逻辑的模型检验将在 16.2.3 小节中介绍,这是一个完整的、典型的模型检验过程;然后将在 16.2.4 小节和 16.2.5 小节中介绍如何利用有序二叉决策树(OBDD)来描述 Kripke 结构及如何进行模型检验,采用该方法可提高对大规模系统的处理能力,优化检验的性能。

16.2.1 Kripke 结构

在模型检验中,一般用 Kripke 结构(Kripke structure)来为反应系统的行为建

模。一个典型的 Kripke 结构一般由一个状态集合、状态转换关系和状态的属性标示函数(用于标示每一个状态中所有为真的属性)组成。Kripke 结构中的路径就对应于系统的计算过程。Kripke 结构的定义如下：

假设 AP 是一个原子命题集合，AP 上的 Kripke 结构 M 是一个四元组，$M=(S,S_0,R,L)$，其中：

(1) S 是一个有限状态集合；

(2) $S_0\subseteq S$，是初始状态集合；

(3) $R\subseteq S\times S$ 是一个 Total 转换关系，即对于任何 $s\in S$，均存在一个状态 s'，使得 $R(s,s')$ 成立；

(4) $L:S\rightarrow 2^{AP}$ 是一个标示函数，标示出每一个状态中所有为真的原子命题。

有时并不关注初始状态 S_0，这样将忽略对其的定义。Kripke 结构 M 中从状态 s 出发的一条路径是一个无限的状态序列 $\pi=s_0s_1s_2\cdots$，其中 $s_0=s$，并且对于所有 $i\geqslant 0$，$R(s_i,s_{i+1})$ 成立。下面是一个 Kripke 结构的实例，本节所有实例数据均来自于文献[5]。

例 16.2.1　对于变量 $x,y\in D=\{0,1\}$，假设一个系统仅有 x,y 两个变量和一个转换关系：$x:=(x+y)\bmod 2$，则 $(x,y)\in D\times D$ 即可描述系统的状态。假设系统的初始状态为 $x=1,y=1$。则其初始状态集合可描述为 $S_0(x,y)\equiv x=1\wedge y=1$。其状态转换关系则可以表示为

$$R((x,y),(x',y'))\equiv x'=(x+y)\bmod 2\ \wedge\ y'=y$$

其中，状态 (x',y') 是 (x,y) 的下一状态。因此，针对该系统建立的 Kripke 结构四元组 $M=(S,S_0,R,L)$ 定义如下：

(1) $S=D\times D$

(2) $S_0=\{(1,1)\}$

(3) $R=\{((1,1),(0,1)),((0,1),(1,1)),((1,0),(1,0)),((0,0),(0,0))\}$

(4) $L((1,1))=\{x=1,y=1\}$，$L((0,1))=\{x=0,y=1\}$，$L((1,0))=\{x=1,y=0\}$，$L((0,0))=\{x=0,y=0\}$

从初始状态存在一条唯一的路径 $(1,1)(0,1)(1,1)(0,1)\cdots$。

16.2.2　计算树逻辑

计算树逻辑(CTL*)公式主要用于描述计算树的属性。而计算树则是通过指定 Kripke 结构中的一个初始状态作为树的根节点，然后将该结构展开成一个无限树而获得。具体可参考图 16.6 中实例，计算树给出了从该初始状态节点出发所有可能的路径。

CTL* 公式由路径量词(path quantifier)和时序操作符(temporal operator)组成。路径量词主要是用于描述计算树的分支结构，包括两种量词：A，表示所有的计算路径；E，表示某一计算路径。路径量词用于限定某一特定状态，表示起始于该状态的所有路径或一些路径满足某些属性。时序操作符主要描述某一特定路经所满足的属性，主要有 5 种基本的操作符：

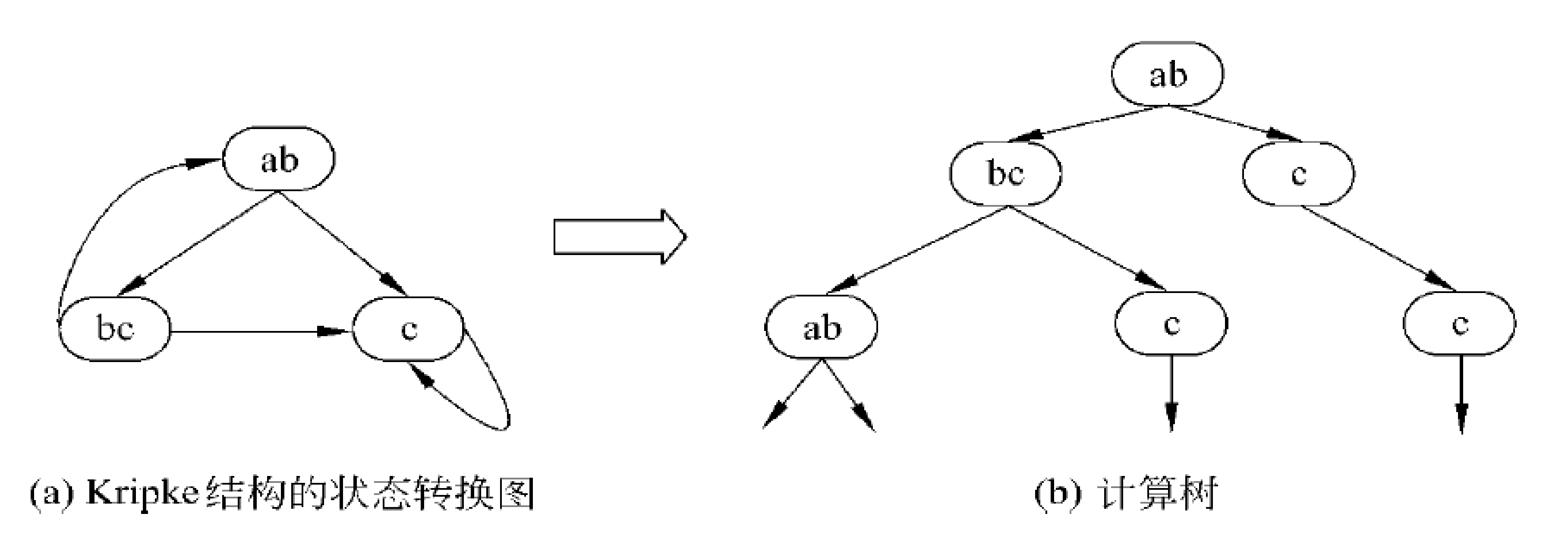


图 16.6 Kripke 结构与计算树

(1) X,"下一次",表示该路径的下一个状态即第二个状态满足相关属性;

(2) F,"最终"或"将来",表示该路径中至少存在一个状态满足相关属性;

(3) G,"总是"或"全部",表示该路径的每一个状态都满足相关属性;

(4) U,"直到",它组合两个属性,表示路径中一直满足第一个属性直到出现了一个状态满足第二个属性;

(5) R,"释放",它也组合了两个属性,表示路径中各状态一直满足第二个属性,直到出现了一个状态满足第一个属性,同时,满足第一个属性的第一个状态同时也满足第二个属性,在此以后,第二个属性不再满足。

在 CTL^* 中,有状态公式(state formula)和路径公式(path formula)两种。假设 AP 是一个原子命题集合。状态公式定义如下:

(1) 如果 $p \in \mathrm{AP}$,则 p 是一个状态公式;

(2) 如果 f 和 g 是状态公式,则 $\neg f$、$f \vee g$ 和 $f \wedge g$ 都是状态公式;

(3) 如果 f 是一个路径公式,则 $\mathrm{E}f$ 和 $\mathrm{A}f$ 是状态公式。

路径公式定义如下:

(1) 如果 f 是一个状态公式,则 f 也是一个路径公式;

(2) 如果 f 和 g 是路径公式,则 $\neg f$、$f \vee g$、$f \wedge g$、$\mathrm{X}f$、$\mathrm{F}f$、$\mathrm{G}f$、$f\mathrm{U}g$、$f\mathrm{R}g$ 均是路径公式。

对于路径 π,用 π^i 表示 π 从 s_i 开始的后缀。假设 f 是一个状态公式,则 $M,s|=f$ 表示 Kripke 结构 M 中的 s 满足状态公式 f。如果 f 是一个路径公式,则 $M,\pi|=f$ 表示 Kripke 结构 M 中的路径 π 满足公式 f。在不会引起误会时,往往将其中的 Kripke 结构 M 省略。假设 f_1 和 f_2 是状态公式,g_1 和 g_2 是路径公式,则:

(1) $M,s|=p \Leftrightarrow p \in L(s)$;

(2) $M,s|=\neg f_1 \Leftrightarrow M,s|\neq f_1$;

(3) $M,s|=f_1 \vee f_2 \Leftrightarrow M,s|=f_1$ 或 $M,s|=f_2$;

(4) $M,s|=f_1 \wedge f_2 \Leftrightarrow M,s|=f_1$ 且 $M,s|=f_2$;

(5) $M,s|=\mathrm{E}g_1 \Leftrightarrow$ 存在一个起始于 s 路径 π,使得 $M,\pi|=g_1$;

(6) $M,s|=\mathrm{A}g_1 \Leftrightarrow$ 任意一个起始于 s 路径 π,均满足 $M,\pi|=g_1$;

(7) $M,\pi|=f_1 \Leftrightarrow s$ 是路径 π 的起始状态,满足 $M,s|=f_1$;

(8) $M,\pi|=\neg g_1 \Leftrightarrow M,\pi|\neq g_1$;

(9) $M,\pi|=g_1\vee g_2 \Leftrightarrow M,\pi|=g_1$ 或 $M,\pi|=g_2$；

(10) $M,\pi|=g_1\wedge g_2 \Leftrightarrow M,\pi|=g_1$ 且 $M,\pi|=g_2$；

(11) $M,\pi|=\mathrm{X}g_1 \Leftrightarrow M,\pi^1|=g_1$；

(12) $M,\pi|=\mathrm{F}g_1 \Leftrightarrow$ 存在一个 $k\geqslant 0$，使得 $M,\pi^k|=g_1$；

(13) $M,\pi|=\mathrm{G}g_1 \Leftrightarrow$ 对于所有 $i\geqslant 0$，均满足 $M,\pi^i|=g_1$；

(14) $M,\pi|=g_1\mathrm{U}g_2 \Leftrightarrow$ 存在 $k\geqslant 0$，均满足 $M,\pi^k|=g_2$；并且对于任意 $0\leqslant j<k$，$M,\pi^j|=g_1$；

(15) $M,\pi|=g_1\mathrm{R}g_2 \Leftrightarrow$ 对于任意 $j\geqslant 0$，$0\leqslant i<j$，如果 $M,\pi^i|\neq g_1$，则 $M,\pi^j|=g_2$。

实际上，通过操作符 ∨、¬、X、U 和 F 可以表达其他 CTL* 公式：

(1) $f\wedge g\equiv\neg(\neg f\vee\neg g)$；

(2) $f\mathrm{R}g\equiv\neg(\neg f\mathrm{U}\neg g)$；

(3) $\mathrm{F}f\equiv \mathrm{True}\,\mathrm{U}f$；

(4) $\mathrm{G}f\equiv\neg\mathrm{F}\neg f$；

(5) $\mathrm{A}f\equiv\neg\mathrm{E}(\neg f)$。

CTL* 有两个子逻辑，一个是分支时间逻辑(branch-time logic)，另一个是线性时间逻辑(linear-time logic)。分支时间逻辑主要用于描述对于一个给定状态的可能路径分支，而线性时间逻辑主要是用于描述单一计算路径中的事件。

CTL 是 CTL* 的子逻辑，其时序操作符 X、F、G、U 和 R 后面必须紧接路径量词。更准确地说，CTL 是在 CTL* 的基础上依据以下规则限定产生：

- 如果 f 和 g 是状态表达式，则 $\mathrm{X}f$、$\mathrm{F}f$、$\mathrm{G}f$、$f\mathrm{U}g$ 和 $f\mathrm{R}g$ 是路径表达式。

  LTL 线性时序逻辑则是在 CTL* 的基础上依据以下规则限定产生：

- 如果 $p\in\mathrm{AP}$，则 p 是一个路径表达式；
- 如果 f 和 g 是路径表达式，则 $\neg f$、$f\vee g$、$f\wedge g$、$\mathrm{X}f$、$\mathrm{F}f$、$\mathrm{G}f$、$f\mathrm{U}g$ 和 $f\mathrm{R}g$ 是路径表达式。

这 3 种逻辑具有不同的表达能力，比如 LTL 中的表达式 $\mathrm{A}(\mathrm{FG}\ p)$ 在 CTL 中无法找到对应的表达方式，同样 CTL 中的表达式 $\mathrm{AG}(\mathrm{EF}\ p)$ 在 LTL 中也没有对应的表达式，而二者的析取 $\mathrm{A}(\mathrm{FG}\ p)\vee\mathrm{AG}(\mathrm{EF}\ p)$ 是 CTL* 表达式，CTL 和 LTL 均无法表达。

后续内容多采用 CTL 描述，以下是 10 个基本的 CTL 操作符：

(1) AX 和 EX；

(2) AF 和 EF；

(3) AG 和 EG；

(4) AU 和 EU；

(5) AR 和 ER。

上述的操作符也可以通过 EX、EG 和 EU 这 3 个操作符表示：

(1) $\mathrm{AX}f=\neg\mathrm{EX}(\neg f)$；

(2) $\mathrm{EF}f=\mathrm{E}[\mathrm{True}\,\mathrm{U}f]$；

(3) $\mathrm{AG}f=\neg\mathrm{EF}(\neg f)$；

(4) $A[fUg]=\neg E[\neg gU(\neg f\wedge\neg g)]\wedge\neg EG\neg g$;

(5) $A[fRg]=\neg E[\neg fU(\neg g)]$;

(6) $E[fRg]=\neg A[\neg fU(\neg g)]$。

在实际的分析验证过程中,往往需要对分析的系统增加一些前提条件,比如在分析协议过程中,可能会假设信道是安全的。这些限定条件在 CTL* 逻辑中可以较好地表达,但在 CTL 逻辑中则需要做适当的修改。这里引入 Fairness 语义。Fairness 约束是状态集合,可能通过逻辑表达式表示。如果 CTL 逻辑中增加了 Fairness 约束,对于约束中的任意一个状态集合,如果路径中至少包含了一个状态属于该集合,则称该路径为 Fairness 路径。

一个 Fair Kripke 结构是一个四元组 $M=(S,R,L,F)$,其中 S、L 和 R 仍如前所定义,$F\subseteq 2^S$,是 Fairness 约束集合。假设 $\pi=s_0,s_1,\cdots$是 M 中的一条路径。定义

$$\inf(\pi)=\{s\mid s=s_i,i\geqslant 0\}$$

则称 π 是 Fair 的,当且仅当对于任意 Fairness 约束 $P\in F$,均有 $\inf(\pi)\cap P\neq\varnothing$。用 $M,s|=_F f$ 表示状态表达式 f 在 Fair Kripke 结构 M 中状态 s 为真,采用 $M,\pi|=_F g$ 表示路径表达式 g 对于 M 中的路径 π 为真。对于前面给出的 15 条公式中,只有 1、5、6 发生变化:

1: $M,s|=_F p\Leftrightarrow$存在一条起始于 s 的 Fair 路径 $p\in L(s)$。

5: $M,s|=_F E(g_1)\Leftrightarrow$存在一条起始于 s 的 Fair 路径 $\pi,\pi|=_F g_1$。

6: $M,s|=_F E(g_1)\Leftrightarrow$对于任意起始于 s 的 Fair 路径 $\pi,\pi|=_F g_1$。

本文中将不对讨论具有 Fairness 约束的模型校验方法,如果读者对相关内容感兴趣,可进一步阅读文献[5]。

16.2.3 CTL 模型校验

模型校验问题利用 CTL* 描述就是,给定一个描述系统的 Kripke 结构 $M=(S,R,L)$和一个逻辑表达式 f 用于描述该系统期望满足的属性,模型校验就是需要在 S 中寻找所有满足该表达式 f 的状态:$\{s\in S|M,s|=f\}$。在这里仅介绍应用最广泛的 CTL 模型检验,LTL 和 CTL* 模型检验可参考文献[5]。

对于 Kripke 结构 $M=(S,R,L)$,如果想知道 S 中哪些状态满足 CTL 表达式 f,将首先对每一个状态 s,表达式的子表达式在状态 s 为真,利用 label(s)进行标记。这里 label(s)初始值就是 $L(s)$。然后将进行一系列的分析步骤,在第 i 步,对第 $i-1$ 步的子命题进行分析,直到结束,这样将可以得到 $M,s|=f$ 当且仅当 $f\in$ label(s)。

任何 CTL 表达式均可利用$\neg$、$\vee$、EX、EU 和 EG 的形式表示。这样中间步骤需要处理 6 种形式的表达式就可以了:$\neg f_1$、$f_1\vee f_2$、EXf_1、$E[f_1Uf_2]$、EGf_1 或者 g 是原子命题。

对于$\neg f_1$,将所有 f_1 没有标记的状态均标记。对于 $f_1\vee f_2$,则将所有 f_1 和 f_2 标记的状态均标记上。对于 EXf_1,如果状态的后继状态被 f_1 标记,则将该状态标记。

对于形如 $g=E[f_1Uf_2]$的表达式,首先搜索所有 f_2 标记的状态,然后通过 R 的逆向操作,查找所有可能到达这些状态的路径,如果这些路径中存在状态被 f_1 标记,

将这些状态标记为 g。

下面给出了 CheckEU 算法，对于任意的状态 s，当且仅当 $s|=f_1$，则 $f_1\in \text{label}(s)$，当且仅当 $s|=f_2$，则 $f_2\in \text{label}(s)$。如果 $f_1,f_2\in \text{label}(s)$，则该函数将 $E[f_1 U f_2]$ 增加到所有满足该表达式状态 s 的标示集合 label(s)中。该算法的计算复杂度为 $O(|S|+|R|)$。

```
Procedure CheckEU(f1,f2)
    T:={s|f2∈label(s)};
    For 所有 s∈T do label(s):=label(s)∪{E[f1Uf2]};
    While T≠∅ do
        选择一个 s∈T,将其从 T中删除;
    For 对于所有满足 R(t,s)的 t do
        If E[f1Uf2]∉label(t),并且 f1∈label(t) then
                label(t):=label(t)∪{E[f1Uf2]};
                T:=T∪{t};
          End if;
      End for;
    End while;
End procedure
```

对于 $g=EGf_1$ 的分析则相对来说更为复杂。它是通过将图分解为非平凡的强连通分量实现的。一个强连通分量(strongly connected component)C 是一个极大子图，其中任意一个节点都可从其他任意的节点通过有向路径到达。C 是非平凡的当且仅当 C 中的节点不只一个节点，或只有一个节点，但该节点存在一个自循环，即存在一条边，起始节点和终止节点均为该节点。

假设 $M=(S,R,L)$，将 S 中 f_1 不为真的状态均删除，得到 S'，然后删除相应转换关系和标示关系，得到 R' 和 L'，这样就得到新的 Kripke 结构 $M'=(S',R',L')$，其中 $S'=\{s\in S|M,s|=f_1\}$，$R'=R|_{S'\times S'}$，$L'=L|_{S'}$。这样，R' 有可能不是闭包的。如果没有任何转换关系，存在的节点将被删除。

引理 16.2.1　$M,s|=EGf_1$ 当且仅当以下两个条件满足：

(1) $s\in S'$；

(2) 在图(S',R')中，存在一个非平凡的强连通分量 C，其中 M' 中存在一条路径从 s 到 C 中的节点 t。

以下算法是根据该引理实现的，该引理的证明可参考文献[5]。

```
Procedure CheckEG(f1)
    S'={s|f1∈label(s)};
    SCC:={C|C是 S'的非平凡强连通部件};
    T:=∪C∈SCC{s|s∈C};
    For 所有 s∈T do label(s):=label(s)∪{EG f1};
    While T≠∅ do
        选择一个 s∈T,将其从 T中删除;
```

```
            For 所有满足 R(t,s)并且 t∈S' do
                If EG f1∉label(t) then
                    label(t):=label(t)∪{E[f1Uf2]};
                    T:=T∪{t};
                End if
            End for
        End while
    End procedure
```

该算法首先生成受约束的 Kripke 结构 $M'=(S',R',L')$，然后将图(S',R')化分成强连通分量。该算法的复杂度是$O(|S'|+|R'|)$。然后，寻找属于非平凡分量的状态，通过转换关系 R'的逆向关系，分析可到达的节点，并且要求到达该节点的路径中各状态均被 f_1 标示。整个计算过程的复杂度是$O(|S|+|R|)$。

定理 16.2.1 对于一个 CTL 表达式 f，判断其在 Kripke 结构 $M=(S,R,L)$中状态 s 时是否为真的时间复杂度是$O(|f|\cdot(|S|+|R|))$。

16.2.4 有序二叉决策图(OBDD)

有序二叉决策图(ordered binary decision diagram)是用于表示布尔公式的重要手段之一，并已成功用于多项计算机辅助设计工作中。在介绍二叉决策图之前，首先了解一下二叉决策树。二叉决策树是一种有根、有向树，包括终端节点和终端节点两种节点。每个非终端节点 v 标示为变量 var(v)，具有两个子节点：low(v)，对应于变量被赋予 0 时的情况；high(v)，对应于变量 v 被赋予 1 时的情况。每一个终端节点 v 被标示为 value(v)，其值是 0 或者 1。

图 16.7 所示是公式 $f(a_1,a_2,b_1,b_2)=(a_1\leftrightarrow b_1)\wedge(a_2\leftrightarrow b_2)$的二叉决策树。

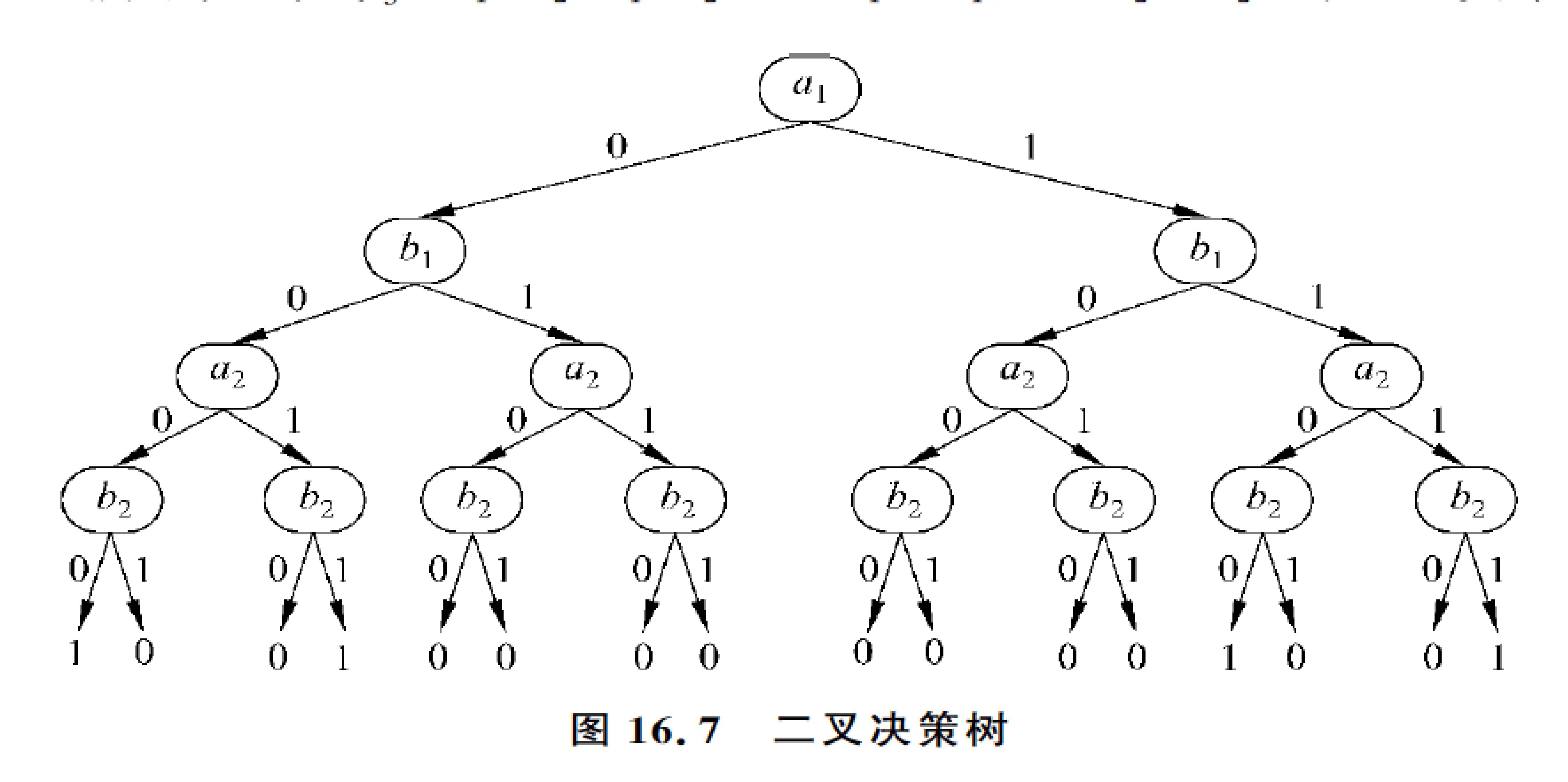


图 16.7 二叉决策树

二叉决策树对于表示布尔函数来说并不是一种简洁的手段，它的大小和真值表一样。但是在二叉决策树中存在大量的冗余，其中存在一些同构的子树，合并这些同构子树可以大大地简化二叉决策树。通过该简化过程，可以得到一个有向非循环图，即二叉决策图。二叉决策图是一种有根、有向非循环图，包括终端节点和非终端节点两种节点。如同二叉决策树的描述，每个非终端节点 v 表示为一个变量 var(v)，具

有两个子节点：low(v)和 high(v)。每个终端节点表示为 0 或者 1。

根为 v 的二叉决策图 B，确定了一个布尔函数 $f_v(x_1, x_2, \cdots, x_n)$，其中：

(1) 如果 v 是一个终端节点：

① 如果 value(v)=1，则 $f_v(x_1, x_2, \cdots, x_n)=1$；

② 如果 value(v)=0，则 $f_v(x_1, x_2, \cdots, x_n)=0$。

(2) 如果 v 是一个非终端节点，并且 var(v)=x_i，则

$$f_v(x_1, x_2, \cdots, x_n) = (\neg x_i \wedge f_{\text{low}(v)}(x_1, x_2, \cdots, x_n)) \vee (x_i \wedge f_{\text{high}(v)}(x_1, x_2, \cdots, x_n))$$

对于两个二叉决策图，如果存在一个节点集合上的一一映射函数 h 使得，对于任意节点终端 v，value(v) = value($h(v)$)；对于任意的非终端节点 v，var(v) = var($h(v)$)，h(low(v))=low($h(v)$)和 h(high(v))=high($h(v)$)。

Bryant 等人提出了一种简化布尔函数表示的方法，在二叉决策图的基础上增加了两条限制：一是各路径中从根节点到终端节点，变量出现的顺序必须一致；另外，在图中不存在同构的子树和冗余的节点。对于第一个限制，只需要在变量之间建立一个序，对于任意的非终端节点 u，如果它存在一个子节点 v，则 var(u)<var(v)。对于第二条限制，则需要通过以下操作实现。

(1) 删除重复的终端节点：对于一个给定的值，0 或者 1，仅保留一个终端节点，删除其他终端节点，将指向被删除节点的边重定向到保留的节点。

(2) 删除重复的非终端节点：如果两个非终端节点 u 和 v，var(u) = var(v)，low(u)=low(v)，并且 high(u)=high(v)，则删除 u，将所有指向节点 u 的边重定向到节点 v。

(3) 删除冗余的测试(test)，如果非终端节点 v，low(v)=high(v)，则删除节点 v，将所有进入边重定向到 low(v)。

通过重复地进行以上操作，直到二叉决策图的大小不再发生变化，就可以得到一个简化的二叉决策图。对于通过以上操作得到的二叉决策图，称之为有序二叉决策图(Ordered Binary Decision Diagram，OBDD)。如果按照 $a_1<b_1<a_2<b_2$ 和 $a_1<a_2<b_1<b_2$ 的顺序，可以得到两个不同的 OBDD，如图 16.8 所示。

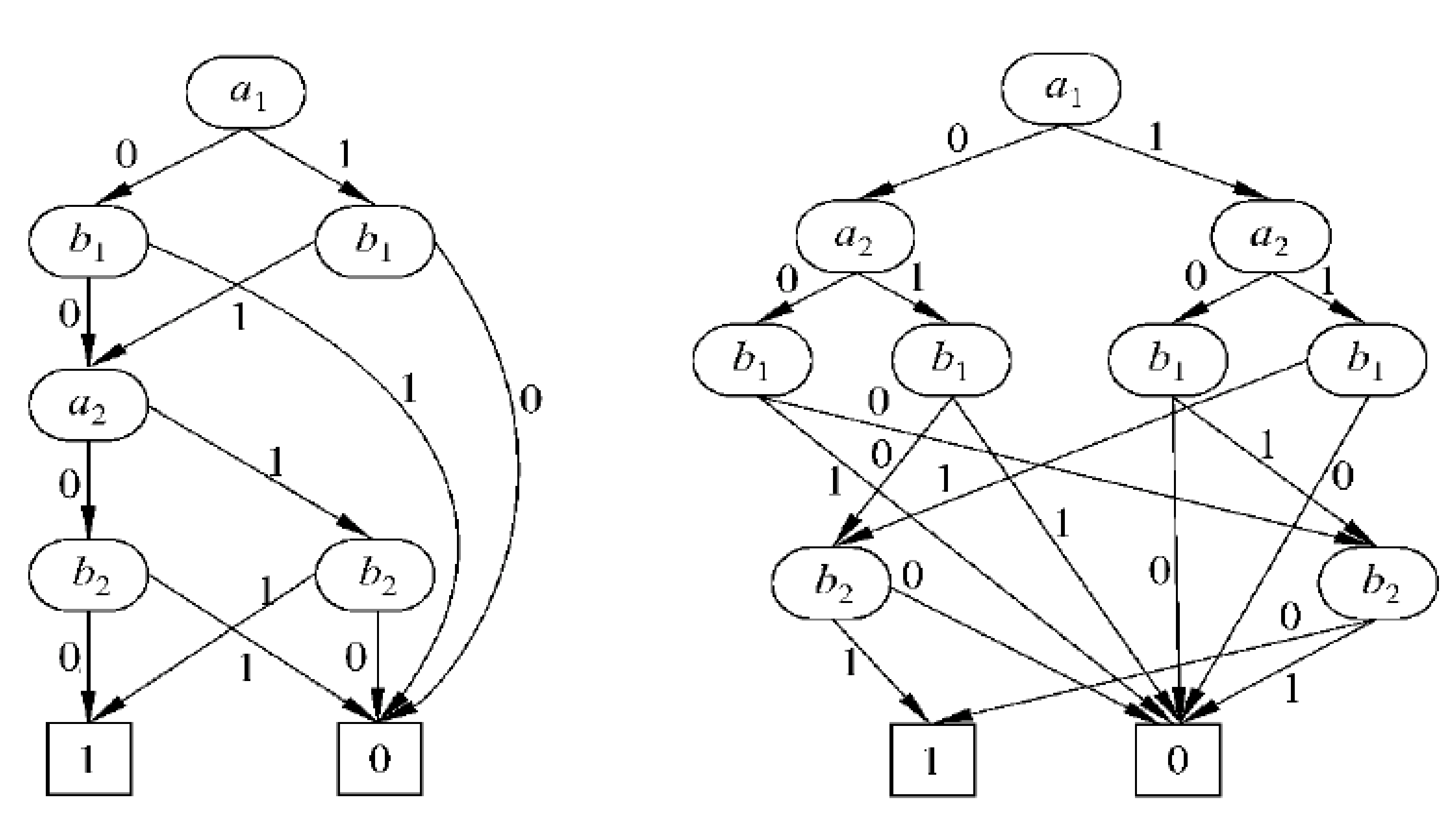


图 16.8　采用不同顺序的有序二叉决策树

从中可以发现，采用不同的顺序对 OBDD 图的大小和复杂程度有严重的影响。针对如何寻找最有效的变量顺序或如何对 OBDD 图中的变量进行重新排序以简化 OBDD 图问题，一些学者提出了各种不同思路。比如针对集成电路应用的深度搜索方式、动态重排(dynamic reordering)技术等。

对于布尔函数 f，当将其中变量 x_i 赋值为 b 时，可以表示为以下形式：

$$f|_{x_i \leftarrow b}(x_1, \cdots, x_n) = f(x_1, \cdots, x_{i-1}, b, x_{i+1}, \cdots, x_n)$$

当 f 表示为 OBDD 时，可以在该 OBDD 基础上进行变换得到 $f|_{x_i \leftarrow b}$的 OBDD 表示。变换采取深度优先的搜索，对于任意一个节点 v，如果它有一条边指向节点 w，而且 $\mathrm{var}(w) = x_i$，则：

(1) 如果 $b=0$，将该边重定向到 $\mathrm{low}(v)$；

(2) 如果 $b=1$，将该边重定向到 $\mathrm{high}(v)$；

然后执行上面的 3 项操作，简化新图，即得到 $f|_{x_i \leftarrow b}$的 OBDD。

其他 16 个双操作数的逻辑操作也可以高效地通过 OBDD 转换实现。其核心思想是利用 Shannon Expansion：$f = (\neg x \wedge f|_{x \leftarrow 0}) \vee (x \wedge f|_{x \leftarrow 1})$。

对于这 16 个操作符，Byrant 给出了一个通用的算法 Apply。先作以下假设：

(1) 假设 $*$ 代表 16 个操作符中的任意一个操作符；

(2) f 和 f'是两个布尔函数，v 和 v'对应于这两个函数的 OBDD 根节点；

(3) $x = \mathrm{var}(v)$和 $x' = \mathrm{var}(v')$；

则有：

① 如果 v 和 v'都是终端节点，则 $f * f' = \mathrm{value}(v) * \mathrm{value}(v')$；

② 如果 $x = x'$，则展开得到：

$$f * f' = (\neg x \wedge (f|_{x \leftarrow 0} * f'|_{x \leftarrow 0})) \vee (x \wedge (f|_{x \leftarrow 1} * f'|_{x \leftarrow 1}))$$

将公式展开成两部分，然后对两部分子公式通过递归逐步解析。对于以上展开的两部分，生成的 OBDD 为一个新的根节点 w，$\mathrm{var}(w) = x$，其 $\mathrm{low}(w)$对应于 $f|_{x \leftarrow 0} * f'|_{x \leftarrow 0}$的 OBDD，$\mathrm{high}(w)$对应于 $f|_{x \leftarrow 1} * f'|_{x \leftarrow 1}$的 OBDD。

③ 如果 $x < x'$，则 f'不依赖于 x，因此，$f'|_{x \leftarrow 0} = f'|_{x \leftarrow 1} = f'$，则

$$f * f' = (\neg x \wedge (f|_{x \leftarrow 0} * f')) \vee (x \wedge (f|_{x \leftarrow 1} * f'))$$

其 OBDD 的构造过程如上所述。

④ 如果 $x' < x$，则其思路与上类似。

下面介绍如何利用 OBDD 来表示 Kripke 结构。首先，假设 Q 是$\{0,1\}$上的一个 n 元关系，这样 Q 可根据其特征函数生成其 OBDD，即

$$f_Q(x_1, \cdots, x_n) = 1 \quad 若 \quad Q(x_1, \cdots, x_n)$$

这样不失一般性，可以进一步假设 Q 是有限域 D 上的 n 元关系，可以假设 D 有 2^m 个元素($m > 1$)。为了利用 OBDD 来描述 Q，需要对 D 中的元素进行编码，$\phi: \{0,1\}^m \to D$。这样，根据该编码 ϕ，可以构造一个新的 $m \times n$ 元布尔关系 $\hat{Q}$：

$$\hat{Q}(\boldsymbol{x}_1, \cdots, \boldsymbol{x}_n) = Q(\phi(\boldsymbol{x}_1), \cdots, \phi(\boldsymbol{x}_n))$$

其中，$\boldsymbol{x}_i$ 是一个 m 维的布尔向量，对应于变量 x_i 的编码。这样 Q 就可以根据 $\hat{Q}$ 的特

征函数 $f_{\hat{Q}}$生成相应的 OBDD。该方法也可以很容易地扩展为基于 $D_1,\cdots,D_n$ 多个不同域的多元关系。

对于 Kripke 结构 $M=(S,R,L)$,需要准确地描述 S、R 和 L。对于 S,假设有 2^m 个状态,可以通过 $\phi:\{0,1\}^m\to S$ 进行映射。对于关系 R,其状态仍然按照 S 的映射方式,关系转换过程仍然通过当前状态 $\boldsymbol{x}$ 和下一步状态 $\boldsymbol{x}'$,即经过转换后得到 $\hat{R}(\boldsymbol{x},\boldsymbol{x}')$,这样 R 的 OBDD 可根据特征函数 $f_{\hat{R}}$得到。

16.2.5　符号模型检验

下面将介绍如何利用 OBDD 描述 Kripke 结构来进行模型检验,该方法之所以称之为"符号模型检验"是因为它的分析过程主要是在布尔表达式的基础上进行的。由于 OBDD 描述的是状态或转换关系的集合,相关操作需要对整个集合而不是独立的状态或转换。这样,引入时序逻辑操作的固定点(fixpoint)概念。集合 $S'\subseteq S$ 是函数 $\tau:\mathcal{P}(S)\to\mathcal{P}(S)$的固定点,当且仅当 $\tau(S')=S'$。

假设 $M=(S,R,L)$是任意一个有限的 Kripke 结构。集合 $\mathcal{P}(S)$是 S 的所有子集,用 $\mathcal{P}(S)$来表示该格。其元素 S'也可以认为是 S 中的一个断言,就是在 S'中的所有状态为真。格中的最小元素是空集,有时也直接采用"False"表示,最大的元素是 S,有时也写为"True"。从 $\mathcal{P}(S)$到 $\mathcal{P}(S)$的映射函数,称之为一个断言转换,假设 $\tau:\mathcal{P}(S)\to\mathcal{P}(S)$,则:

(1) 如果 $P\subseteq Q$ 蕴含着 $\tau(P)\subseteq\tau(Q)$,则称 τ 是单调的;

(2) 如果 $P_1\subseteq P_2\subseteq\cdots$蕴含着 $\tau(\bigcup_i P_i)\subseteq\bigcup_i\tau(P_i)$,则称 τ 是$\bigcup$连续的;

(3) 如果 $P_1\supseteq P_2\supseteq\cdots$蕴含着 $\tau(\bigcap_i P_i)\subseteq\bigcap_i\tau(P_i)$,则称 τ 是$\bigcap$连续的。

用 $\tau^i(Z)$表示在集合 Z 上应用 τ 函数 i 次,其中 $\tau^0(Z)=Z,\tau^{i+1}(Z)=\tau(\tau^i(Z))$。$\mathcal{P}(S)$中的单调的断言转换 τ 永远存在一个最小固定点(least fixpoint),$\mu Z.\tau(Z)$和一个最大固定点(greatest fixpoint),$\nu Z.\tau(Z)$。

(1) 如果 τ 是单调的,则 $\mu Z.\tau(Z)=\bigcap\{Z|\tau(Z)\subseteq Z\}$,$\nu Z.\tau(Z)=\bigcup\{Z|\tau(Z)\supseteq Z\}$。

(2) 如果 τ 是$\bigcup$连续的,则 $\mu Z.\tau(Z)=\bigcup_i(\tau^i(\text{False}))$。

(3) 如果 τ 是$\bigcap$连续的,则 $\nu Z.\tau(Z)=\bigcap_i\tau^i(\text{True})$。

引理 16.2.2　如果 S 是有限的,τ 是单调的,则 τ 是$\bigcup$连续的和$\bigcap$连续的。

引理 16.2.3　如果 τ 是单调的,则对于任意的 i,$\tau^i(\text{False})\subseteq\tau^{i+1}(\text{False})$,并且 $\tau^i(\text{True})\supseteq\tau^{i+1}(\text{True})$。

引理 16.2.4　如果 S 是有限的,τ 是单调的,则存在一个整数 i_0,使得任意 $j\geqslant i_0$,$\tau^j(\text{False})=\tau^{i_0}(\text{False})$,同样,也存在一个整数 j_0,使得任意 $j\geqslant i_0$,$\tau^j(\text{True})=\tau^{j_0}(\text{True})$。

引理 16.2.5　如果 S 是有限的,τ 是单调的,则存在一个整数 i_0,使得 $\mu Z.\tau(Z)=\tau^{i_0}(\text{False})$;也存在一个整数 j_0,使得 $\nu Z.\tau(Z)=\tau^{j_0}(\text{True})$。

```
function Lfp(Tau: PredicateTransformer): Predicate
    Q: =False;
    Q'=Tau(Q);
    while(Q≠Q')do
```

```
            Q:=Q';
            Q':=Tau(Q');
        end while
        return(Q);
    end function

    function Gfp(Tau: PredicateTransformer): Predicate
        Q:=True;
        Q'=Tau(Q);
        while (Q≠Q') do
            Q:=Q';
            Q':=Tau(Q');
        end while
        return(Q);
    end function
```

在 $\mathscr{D}(S)$中的断言$\{s|M,s|=f\}$基础上分析 CTL 表达式 f 时，每一个 CTL 操作可以转换为类似函数的最小固定点和最大固定点分析：

(1) $\mathrm{AF}f_1=\mu Z.f_1\vee \mathrm{AX}Z$；

(2) $\mathrm{EF}f_1=\mu Z.f_1\vee \mathrm{EX}Z$；

(3) $\mathrm{AG}f_1=v Z.f_1\wedge \mathrm{AX}Z$；

(4) $\mathrm{EG}f_1=v Z.f_1\wedge \mathrm{EX}Z$；

(5) $\mathrm{A}[f_1\mathrm{U}f_2]=\mu Z.f_2\vee(f_1\wedge \mathrm{AX}Z)$；

(6) $\mathrm{E}[f_1\mathrm{U}f_2]=\mu Z.f_2\vee(f_1\wedge \mathrm{EX}Z)$；

(7) $\mathrm{A}[f_1\mathrm{R}f_2]=v Z.f_2\wedge(f_1\vee \mathrm{AX}Z)$；

(8) $\mathrm{E}[f_1\mathrm{R}f_2]=v Z.f_2\wedge(f_1\vee \mathrm{EX}Z)$。

直观上讲，最小固定点对应于“最终会出现的属性”，最大固定点对应于“一直具备的属性”。下面将证明 EG 和 EU 的固定点特性。

引理 16.2.6 $\tau(Z)=f_1\wedge \mathrm{EX}Z$ 是单调的。

引理 16.2.7 假设 $\tau(Z)=f_1\wedge \mathrm{EX}Z$，且 $\tau^{i_0}(Z)$是序列 $\mathrm{True}\supseteq\tau(\mathrm{True})\supseteq\cdots$的极限。对于任意 $s\in\tau^{i_0}(\mathrm{True})$，则 $s|=f_1$，并且存在一个状态 s'，使得$(s,s')\in R$，并且 $s'\in\tau^{i_0}(\mathrm{True})$。

引理 16.2.8 $\mathrm{EG}f_1$ 是函数 $\tau(Z)=f_1\wedge \mathrm{EX}Z$ 的一个固定点。

引理 16.2.9 $\mathrm{EG}f_1$ 是函数 $\tau(Z)=f_1\wedge \mathrm{EX}Z$ 的一个最大固定点。

引理 16.2.10 $\mathrm{E}[f_1\mathrm{U}f_2]$是函数 $\mathrm{E}[f_1\mathrm{U}f_2]=f_2\vee(f_1\wedge \mathrm{EX}Z)$的最小固定点。

在 16.2.4 小节中，介绍了 CTL 模型检验算法，该算法的复杂度和图的大小及表达式的长度是线性相关的，应该说速度已经相当快了。但随着分析目标越来越复杂、越来越庞大，CTL 检验仍存在状态空间爆炸的问题，下面将介绍利用 OBDD 来描述 Kripke 结构，进而在其基础上进行模型检验，将可提高其对状态的描述能力，在一定程度上解决状态空间爆炸问题。在此之前，首先介绍量化布尔表达式 QBF (Quantified Boolean Formulas)。

对于一个命题变元集合 $V=\{v_0,\cdots,v_{n-1}\}$，QBF(V)是满足以下条件的最小集合：

(1) V 中的每个变元都是表达式；

(2) 如果 f 和 g 是表达式，则 $\neg f$、$f \vee g$ 和 $f \wedge g$ 是表达式；

(3) 如果 f 是表达式，并且 $v \in V$，则 $\exists v f$ 和 $\forall v f$ 均是表达式。

对于 QBF(V)的真值赋值是一个函数 $\sigma: V \to \{0,1\}$。假设 $a \in \{0,1\}$，则用以下方式表示真值赋值 $\sigma<v \leftarrow a>$，其定义如下：

$$\sigma < v \leftarrow a > = \begin{cases} a & v = w \\ \sigma(w) & v \neq w \end{cases}$$

如果 f 是 QBF(V)中的一个表达式，σ 是一个真值赋值，$\sigma|=f$ 表示在赋值 σ 的情况下，f 为真。因此：

(1) $\sigma|=v$ 若 $\sigma(v)=1$；

(2) $\sigma|=\neg f$ 若 $\sigma|\neq f$；

(3) $\sigma|=f \vee g$ 若 $(\sigma|=f) \vee (\sigma|=g)$；

(4) $\sigma|=f \wedge g$ 若 $(\sigma|=f) \wedge (\sigma|=g)$；

(5) $\sigma|=\exists v f$ 若 $(\sigma<v\leftarrow 0>|=f) \vee (\sigma<v\leftarrow 1>|=f)$；

(6) $\sigma|=\forall v f$ 若 $(\sigma<v\leftarrow 0>|=f) \wedge (\sigma<v\leftarrow 1>|=f)$。

QBF 表达式和普通的命题表达式具有同样的表达能力，但在符号模型检验过程中使用更方便。每一个 QBF 表达式确定了一个集合 V 上的 n 元布尔关系。前面曾经介绍了如何将一个命题逻辑转换成 OBDD 方式。QBF 中的量词可以转换成以下组合表达方式：

(1) $\exists x f = f|_{x\leftarrow 0} \vee f|_{x\leftarrow 1}$；

(2) $\forall x f = f|_{x\leftarrow 0} \wedge f|_{x\leftarrow 1}$。

符号模型检验的算法主要通过一个 Check 函数实现，该函数的输入参数是代检验的 CTL 表达式，返回的是使该表达式为真的状态的 OBDD 描述。当然，其返回值也依赖于被检测系统的转换关系的 OBDD 描述。

如果 f 是一个原子命题 a，则 Check(f)是满足命题 a 的状态的 OBDD 描述。

如果 $f=f_1 \wedge f_2$ 或者 $f=\neg f_1$，则 Check(f)通过 16.2.5 小节中介绍的 Apply 算法获得。

对于 EXf，E[fUg]和 EGf，则通过以下方式处理：

(1) Check(EXf)=CheckEX(Check(f))

(2) Check(E[fUg])=CheckEU(Check(f),Check(g))

(3) Check(EGf)=CheckEG(Check(f))

其中，CheckEX、CheckEU、CheckEG 参数都是 OBDD 的形式，而 Check 参数是 CTL 表达式的形式。由于所有的时序逻辑操作符都可以通过以上的形式表达，所以以上的定义涵盖了所有的时序逻辑操作符。

对于 CheckEX，EXf 在状态 s 为真，当且仅当其存在一个后续状态 s' 中 f 为真，即

$$\text{CheckEX}(f(\boldsymbol{v})) = \exists \boldsymbol{v}'[f(\boldsymbol{v}') \wedge R(\boldsymbol{v},\boldsymbol{v}')]$$

其中，$R(\boldsymbol{v},\boldsymbol{v}')$是 OBDD 中的转换关系。如果有 f 和 R 的 OBDD，则可以通过 QBF 表达式方式生成表达式 $\exists \boldsymbol{v}'[f(\boldsymbol{v}') \wedge R(\boldsymbol{v},\boldsymbol{v}')]$的 OBDD。

CheckEU 是基于最小固定点的特性来完成的。

$$\mathrm{E}[f_1 \mathrm{U} f_2] = \mu Z.\, f_2 \vee (f_1 \wedge \mathrm{EX} \quad Z)$$

当利用算法 Lfp 计算时，将得到以下序列：

$Q_0, Q_1, \cdots, Q_i, \cdots$

如果有 f、g 和当前 Q_i 的 OBDD，可以很容易地获得 Q_{i+1}。OBDD 提供了规范的布尔函数表示方式，很容易比较二者的重叠程度。当 $Q_i = Q_{i+1}$ 时，Lfp 算法终止，得到的 Q_i 就是对应于 $\mathrm{E}[f_1 \mathrm{U} f_2]$的状态的 OBDD。

CheckEG 和 CheckEU 类似，是基于最大固定节点特性，即

$$\mathrm{EG} f_1 = \nu Z.\, f_1 \wedge \mathrm{EX} \quad Z$$

根据以上介绍的对各种典型表达式的检验方法，可以将一个大的表达式拆成不同的子表达式。

以上仅仅介绍了模型校验最基础的思想和方法，关于模型校验的研究相当广泛，仍有很多问题还处于研究阶段中，如根据源程序自动构造相应的描述模型等。在文献[6]中对模型校验的问题作了较为系统和深入的介绍，感兴趣的读者可以仔细阅读，一些研究动态在该书作者 Edmund M. Clarke 的个人主页中也有大量介绍，读者可随时关注。

16.3 动态污点传播

动态污点传播分析(dynamic taint propagation analysis)主要是在程序运行过程中对特定的数据进行追踪，是一种典型的动态数据流分析方法。目前被广泛应用于程序的安全性分析、漏洞发掘和恶意代码行为分析之中。

动态污点传播技术是通过对敏感信息进行标记，跟踪记录污点数据被可执行代码引用和执行的情况，从而得到代码对污点数据的操作流程。分析过程中，可以根据预先设定的规则判断污点数据的使用方式是否合法，并获得污点数据操作的细节。污点传播技术可以提供有关污点数据执行的大量真实可靠的信息，因此基于动态污点传播技术可以构造功能强大、性能优越的代码分析和实时监控系统。

在接下来的部分，将按以下方式介绍动态污点传播技术。首先，归纳总结动态污点技术的特点和优势；然后，介绍基于动态污点传播的分析系统的基本组成和工作原理，对主要的技术环节进行分析；最后，将讨论基于动态传播的分析系统的实现方式，并分析系统实现需要解决的关键问题。

基于动态污点传播技术的代码分析能够根据用户的关注点，高效率地获取真实、可靠的代码执行的细节信息，并且不依赖于源代码，因而具有很好的分析性能和扩展潜力，并有着广阔的应用前景。具体而言，污点传播技术的代码分析具有以下的特点。

第一，动态污点传播技术基于 CPU 执行指令对代码进行分析，通过操作 CPU

指令，可以跟踪获取大量细节信息，提供足够精细的分析粒度。

第二，基于指令的这个特征也使动态污点传播分析方法不依赖软件源代码，立足于底层，比分析对象有更高的权限，因此分析所受的干扰小，获取信息真实可靠，分析准确性高，可以有效降低漏报率和误报率。

第三，基于污点传播的分析可以通过对污点源、传播方式及判定规则的设计，进行有针对性的分析，分析任务多样化，分析粒度可控，再加上可以获取足够的代码执行细节信息，因此基于污点传播的分析检测目标明确，分析结论直观。

第四，动态污点传播技术可以在攻击发生之前检测到攻击行为，并且可以检测到未知攻击行为，再结合它的其他特性，动态污点传播技术很适合作为实时监控和主动防护系统的基础技术。

最后，基于污点传播的分析由于可以获得污点传播过程记录和细节信息，使得分析结果的可重现性很强，可靠性高，可以作为语义分析法提取攻击特征的素材，在自动提取攻击特征方面具有特别的优势。

16.3.1 基本原理

一个基于动态污点传播的分析系统通常由污点标记引擎、传播跟踪引擎、规则判断引擎 3 大基本功能模块组成，分别对污点源进行识别和标记、跟踪记录污点数据的操作和传播、判断污点数据的使用是否违背预先设定的规则。除此之外，由于动态污点传播系统本身具有很强的可扩展性，根据功能需要，有的系统还有后续的恶意代码特征提取模块（如 TaintCheck[7]），有的系统则加入了代码和程序映射模块等（如 Panorama[8]）。

下面详细说明动态污点分析系统的 3 大基本功能模块，并分析它们的工作原理。

1. 污点标记

污点标记模块对指定的程序数据进行污点标记。本质上是为程序的数据关联一个标记空间（内存地址）。如果这个数据需要记为污点，它的标记空间将会被置位。如果一个动态污点分析系统需要更多的关联信息，那么污点数据的标记空间将指向一个描述当前污点信息的结构体。对于复杂的系统，可以采用多种不同的污点标记，对各种污点源区别标识，以求同时实现更复杂的分析流程。

为了分析的灵活性和准确性，污点标记模块需要具有对多种污点源进行标记的能力。理想的污点标记模块能够根据用户的设计，识别出感兴趣的污点源并进行标记。对于基于动态污点传播的代码分析系统，经常可能使用的污点源包括以下几种：特定函数变量和特定的内存地址、特定函数的返回值、某一类的 IO 流、特定的 IO 流。

2. 污点传播

污点传播模块是动态污点分析系统的核心，它追踪记录污点数据在程序中的操作，判断是否感染新的数据，标记新的污点数据，并继续对新的污点数据进行追踪。

污点传播策略是判断相关数据是否被污染的关键，直接关系到传播流程的连续性、完整性和可靠性，并最终影响到后续判断的准确性，因此污点传播策略对整个动

态污点分析系统的性能有至关重要的影响。总结已有的关于污点传播的工作，目前受到重视的污点传播方式有以下一些。

（1）直接的污点传播

直接的污点传播是污点数据最直观、明显的传播感染方式，也是最常用的传播方式，可以分为以下两类。

第一类是通过数据移动指令的传播。这类指令包括 LOAD、STORE、MOVE、PUSH、POP 等。如果源操作数是污点数据，那么目标操作数就会被感染，成为新的污点数据。

第二类是通过算术指令的传播。这类指令包括 ADD、SUB、XOR 等。如果源操作数中存在污点数据，那么操作结果就会被感染，成为新的污点数据。

需要说明的是，由算术指令构成的常值函数会对污点传播造成一定的影响。比如 XOR ax，ax 语句，无论 ax 原来的输入值是多少，最终结果都为零。因此，理论上该语句不应引起污点传播，即便输入的是污点数据。但是常值函数的构造可以是多种多样的，要完全识别常值函数并制定特殊的传播策略是非常困难的。有研究表明，即便忽略常值函数的存在，只会使分析更趋于“保守”，实际并不会引起特别的问题。另外，对含有地址产生的语句（如 movl ＄0xa，－12（％ebp））引起的传播也有不同的策略，有的系统只考虑该内存地址的数据本身，而忽略产生地址的数据是否是污点数据，理由是污点数据作为地址偏移量是普遍存在的。这样的处理会造成什么影响还没有明确的结论。

（2）间接的污点传播

间接的污点传播是污点数据的一种并不直观的感染影响其他数据的方式。和与基于数据操作相关的直接传播相比，间接传播往往与条件控制语句相关，属于控制依赖的传播方式。间接的污点传播有两种主要形式。

第一种是条件控制形式的传播。污点数据在条件判断语句中发挥作用，影响其他数据的值，从而达到污点传播的效果。比如下面的代码：

```
switch(x) {
    case 'a': y='a'; break; case 'b': y='b'; break; …
}
```

实现了 x 对 y 的赋值，但是仅仅通过直接的污点传播策略，y 并不会被污点数据 x 感染。另外，条件控制传播通常还会考虑 FLAG 寄存器在污点传播中的作用，这也是基于数据的直接传播方式通常忽略的。

第二种是“查找表”形式的传播。这种情况下，污点输入可能作为“索引值”查找相应的表项，再赋值给某个变量。这样污点数据影响了该变量的值，然而仅仅通过直接的污点传播，受污染的索引值并不会感染其他变量。事实上，查找表形式的传播也可以比较容易地转换为语义上等效的条件控制形式的传播，属于条件控制形式传播的一种特殊实现。

利用控制流图的术语，可以描述一般形式的条件控制传播的算法如下[9]：

(1) 当执行到一个条件分支语句 br 时，计算 br 的所有源操作数的污点标记集合为 taint，并向集合 S 添加<br, taint>项；

(2) 当执行到 br 的直接后控制节点是，删除集合 S 中所有的<br, y>形式的项；

(3) 当执行到任意语句 st 时，如果集合 S 不为空，则把 S 中每一项的 taint 添加到 st 的目标操作数的 taint 中去。

虽然间接的污点传播方式比较隐蔽，实现起来也比较复杂，但是有研究表明，间接的基于条件控制的污点传播是普遍存在的，如果忽略这类传播，将会明显地减少污点传播感染的数据，可能对分析结果造成影响。

3. 规则判断

规则判断模块检查污点数据是否被非法使用，并根据非法使用具体违背的规则，从而判断恶意操作的性质以及软件漏洞的详情。

当使用动态污点分析系统进行代码分析时，结合具体的分析目标，可以设定特定的判断规则，使得分析结果更加具有针对性。在利用动态污点分析系统作为漏洞发掘工具或者实时监控系统时，往往会根据常见的恶意代码对污点数据的非法操作特性，使用归纳总结出的通用规则进行判断。从这个意义上讲，动态污点传播系统似乎只能应对已知的恶意行为。但事实上，由于动态污点传播系统的分析对象已经很靠近底层，几乎都是对各种恶意攻击行为最根本、最本质的一环进行审计，因此所使用的判断规则的归纳性和覆盖性很强。实验表明，少数几条通用规则足以覆盖几乎所有的恶意攻击行为。

用来判断污点数据非法使用的常见的通用规则包括以下几种。

(1) 污点数据用作跳转地址。这是污点数据最常见的一种非法使用。跳转地址包括函数返回地址、函数指针、函数指针的偏移量。这种攻击行为往往是为了将代码执行流程重定向，转到攻击者代码、特权库函数或程序的其他地址（比如用以绕过安全检查部分）。正常情况下，污点数据不会被用来作为跳转对象，因此这条判断规则的误报率很低。

(2) 污点数据用作格式字符串。这是针对格式字符串攻击的一条检测规则。

(3) 污点数据用作系统调用参数。根据需要可以利用这条规则保护某些敏感系统调用的参数不被污点数据篡改。

(4) 污点数据用作特定函数或库函数的输入。和上面一条规则类似，也是为了防止已知不应使用污点数据作为输入参量的函数的输入被污点数据所篡改。

最后，需要特别说明间接的污点传播的判定规则问题。如前所述，间接的污点传播有它的合理性，但正如实验表明的那样，使用污点数据影响条件控制在正常软件中是普遍合法存在的，引入间接的污点传播后会使污点数据成倍扩张，有可能使得本来合法的使用也违背某些通用的判定规则，引起误报发生。因此，有的动态污点传播系统禁用了间接的污点传播策略，而引入间接的污点传播后如何制定具有区分度的判定规则目前还没有结论。

16.3.2 系统的实现

动态污点分析系统需要对 CPU 执行指令进行操作，因此通常借助虚拟机环境或者专门的 DBI(Dynamic Binary Instrument)工具，构造相应的功能模块，对执行指令进行提取、标记、分析。为了便于改造，往往使用开源的系统，如 qemu 虚拟机或 Valgrind DBI 工具。

以基于 Valgrind 的 TaintCheck[7] 的实现为例。首先，Valgrind 采用“反汇编-重组”的工作流程，把 X86 指令(二进制代码)分块翻译成它自己的一套 RISC 类似的指令集，称为 UCode 块。TaintCheck 在 UCode 块上嵌入 taint 分析代码，然后将改好的 UCode 块传回给 Valgrind，Valgrind 再将 UCode 块译回 X86 指令，并在自己的环境中执行。已经做过处理的 UCode 块会保留在 Valgrind 的缓存中，避免多次使用时重复转换的问题。Panarama 等污点传播分析系统也大致采用了类似的结构。

下面将说明实现动态污点分析系统的关键环节——影子内存(shadow memery)和目前已经实现的动态污点分析系统的性能和局限性。

1. *影子内存*

动态污点分析系统最为核心的要求是实现对“影子内存”的支持，这也是动态污点分析系统结构上最大的特点。在动态污点分析系统中，执行代码内存空间的每个字节都被分配一段对应的影子内存，记录污点标记。影子内存的大小根据需要决定，完善的分析系统的影子内存通常具有多层次的组织结构，甚至有类似页表的机制以扩展空间。以 TaintCheck 为例，每个数据(内存中的每个字节)，包括寄存器、栈、堆等，都配有一个 4 字节的影子内存，如果数据被污染，则向它的影子内存存入一个指针，指向一个为它分配的记录 taint 信息的数据结构；否则，影子内存为空指针。从某种意义上讲，污点分析系统的运行实质上就是产生并维护代码执行空间和污点记录空间之间的映射关系。

在对影子内存的功能支持方面，虚拟机环境具有天然的优势。重型的 DBI 工具如 Valgrind 也实现了影子内存的功能，称为 value shadow。参考文献[10]中详细讨论了 DBI 工具实现影子内存支持需要满足的要求。

2. *性能和局限*

动态污点分析系统往往基于虚拟机环境或者专门的 DBI 工具实现，不可避免地造成代码执行效率的下降。一个应用如果运行在动态污点分析系统之中，实验表明使用基于 Valgrind 的 TaintCheck 会有 6～40 倍的性能下降，而使用基于 Qemu 的 Panorama 时性能下降在 20 倍左右。对于代码分析而言，这是可以接受的。对于使用动态污点传播的主动防护系统，这样的性能下降会对应用造成一些影响。不过实验表明对于具有 IO 性能约束的应用，这样的代码执行性能仍然具有实用性。

基于虚拟机环境或者专门的 DBI 工具实现动态污点分析系统，另一方面也带来分析方面的一定的局限性。比如，改造自虚拟机的污点分析系统，往往会继承基于虚拟机的代码分析的常见缺陷，比如恶意代码可能检测运行环境并选择执行流程，不表

现出恶意行为,从而导致检测失败;恶意代码甚至可以利用虚拟机自身的漏洞展开攻击,使虚拟机系统崩溃,致使分析无法进行。基于 DBI 工具的系统实现,同样可能面临类似的问题。

16.4 注记

当前软件分析相关研究比较活跃,涉及多种基础理论与技术,本章仅介绍了几种典型的方法与技术。符号执行、软件测试等方法与技术在软件分析中也得到广泛应用,感兴趣的读者可参阅文献[11,12]。

参 考 文 献

[1] Weiser M. Program Slicing, IEEE Transations on Software Engineering, 10 (4), 1984

[2] 李必信. 程序切片技术及其应用. 北京: 科学出版社,2006

[3] Horwitz S, Reps T, Binkley D. Interprocedural Slicing Using Dependence Graphs, ACM Transactions on Programming Languages and Systems (TOPLAS), 12(1), January,1990

[4] Agrawal H, Horgan J R. Dynamic Program Slicing, Proceedings of the ACM SIGPLAN'90 Conference on Programming Language Design and Implementation, 246-256,1990

[5] Canfora G, Cimitile A, A De Lucia. Conditioned program slicing, Information and Software Technology Special Issue on Program Slicing, 1998

[6] Edmund M, Clarke Jr., Orna Grumberg, Doron A. Peled, Model Checking, The MIT Press,1999

[7] Newsome J, Song D. Dynamic Taint Analysis for Automatic Detection, Analysis, and Signature Generation of Exploits on Commodity Software, Proc. 12th Ann. Network and Distributed System Security Symp. (NDSS'05), Feb. 2005

[8] Heng Yin, Dawn Song, Manuel Egele, Christopher Kruegel, Engin Kirda. Panorama Capturing System-wide Information Flow for Malware Detection and Analysis, 14th ACM Conference on Computer and Communications Security, Alexandria, VA, November 2007

[9] James Clause, Wanchun Li, Alessandro Orso. Dytan: A Generic Dynamic Taint Analysis Framework, ISSTA'07, July 9-12, 2007, London, England, United Kingdom

[10] Nicholas Nethercote, Julian Seward, Valgrind A Framework for Heavyweight Dynamic Binary Instrumentation, ACM SIGPLAN Notice, Volume 42 , Issue 6, Proceedings of the 2007 PLDI conference,Pages: 89-100,2007

[11] James C. King,Symbolic execution and program testing, Communications of the ACM, Volume 19, Issue 7, July 1976

[12] Edvardsson J. A survey on automatic test data generation. In Proceedings of the Second Conference on Computer Science and Engineering in Link¨oping, pages 21-28. ECSEL, October 1999

读者意见反馈

亲爱的读者：

感谢您一直以来对清华版计算机教材的支持和爱护。为了今后为您提供更优秀的教材，请您抽出宝贵的时间来填写下面的意见反馈表，以便我们更好地对本教材做进一步改进。同时如果您在使用本教材的过程中遇到了什么问题，或者有什么好的建议，也请您来信告诉我们。

地址：北京市海淀区双清路学研大厦 A 座 602 室 计算机与信息分社营销室 收
邮编：100084　　　　电子邮件：jsjjc@tup.tsinghua.edu.cn
电话：010-62770175-4608/4409　　　　邮购电话：010-62786544

教材名称：信息安全中的数学方法与技术
ISBN：978-7-302-20966-9

个人资料

姓名：________ 年龄：______ 所在院校/专业：____________
文化程度：________ 通信地址：____________________
联系电话：________ 电子信箱：____________________

您使用本书是作为： □指定教材 □选用教材 □辅导教材 □自学教材

您对本书封面设计的满意度：
□很满意 □满意 □一般 □不满意　改进建议________________

您对本书印刷质量的满意度：
□很满意 □满意 □一般 □不满意　改进建议________________

您对本书的总体满意度：
从语言质量角度看 □很满意 □满意 □一般 □不满意
从科技含量角度看 □很满意 □满意 □一般 □不满意

本书最令您满意的是：
□指导明确 □内容充实 □讲解详尽 □实例丰富

您认为本书在哪些地方应进行修改？（可附页）

____

____

您希望本书在哪些方面进行改进？（可附页）

____

____

电子教案支持

敬爱的教师：

为了配合本课程的教学需要，本教材配有配套的电子教案（素材），有需求的教师可以与我们联系，我们将向使用本教材进行教学的教师免费赠送电子教案（素材），希望有助于教学活动的开展。相关信息请拨打电话 010-62776969 或发送电子邮件至 jsjjc@tup.tsinghua.edu.cn 咨询，也可以到清华大学出版社主页（http://www.tup.com.cn 或 http://www.tup.tsinghua.edu.cn）上查询。